1/86

CPU bringt

6,– DM
6,– sFr
50,– öS
18,– FMK

Spezial-Doppelausgabe Jan./Febr.

1. Programm für: Atari 520 ST

TIPS + Tricks

- Checksummen für C-64
- Spectrum Assembler-Kurs
- Serie: universelles Bus-System
- Komfortabler Makro-Assembler für C-64
- Toolkit, CPC
- Sektor-Kopierer

Softwaretest: Turbo-Pascal 3.0 für CPC 464

```
on CP/M 80 for CPC 464   (c) F.Brall

ty fuer Turbo-Pascal Programmierer *)
tei auf Epson Mx80TypIII ausgeben *)
wird der Original ASCII Zeichensatz  *)
mit allen Sonderzeichen / *)
(* Der Deutsche Printer-Zeichensatz muss nicht abgeschaltet werden *)
(* Autor Frank Brall 13.10.85 *)
name     :string20;
t        :text;
zeichen  :char;
ok       :boolean;
edure bit8on;
egin
write(lst,chr(27),chr(62));
d;
edure bit8off;
in
rite(lst,chr(27),chr(61));
end;
egin
clrscr;
writeln(' **** LIST ****        Copyright Frank Brall  1985');
writeln;
writeln;
write ('Filenamen ? ');
readln (name);
assign (t,name);
(*$i-*) reset (t); (*$i+*)
ok := ioresult=0;
if not ok then writeln('Datei ist nicht auf Diskette !');
if ok then
 begin
  writeln;
  write('<f> Fettdruck  <n> Normaldruck  ?');
  read(zeichen);
  if zeichen = 'f' then write(lst,chr(27),chr(69));
  if zeichen = 'n' then write(lst,chr(27),chr(70));
  writeln(lst,'Turbo 3.0 Version on CP/M 80 for CPC 464   (c) F.Brall');
  writeln(lst);
  writeln(lst);
  while not eof(t) do
   begin
    read(t,zeichen);
    case zeichen of
     '':zeichen := chr(130);
     '':zeichen := chr(132);
     '':zeichen := chr(131);
     '':zeichen := chr(133);
     '(*':begin write(lst,'('); zeichen := '*' end;
     '*)':begin write(lst,'*'); zeichen := ')' end;
    nd (* case *);
    if ord(zeichen) > 128 then
     begin
      zeichen := chr(ord(zeichen)-128);
      bit8on;
     end
     else  bit8off;
    write(lst,zeichen);
    write (zeichen);
   end (* while eof *);
  end (* ok *);
  close(t);
end.
```

Top-Programme

„Entschnüren“ Sie doch unsere Programmpakete!

„Herr Müller, eigentlich wollten Sie mir ja die tollen Anwender-Programm-Pakete zeigen!“

- Super-Kombi-Discs
- Super-Games
- Utility Special
- Best of . . . Discs
- Apple-Utility
- Atari-Utility

„TIPS & TRICKS von CPU“ hält für Sie ein einmalig günstiges Angebot bereit: Die Redaktion hat sechs tolle Programm-Pakete ausgesucht und versucht Sie nun „loszuwerden“! Über die Preise reden wir jetzt einmal nicht. Diese sind so günstig, daß Sie sie uns doch nicht abnehmen würden. A propos „abnehmen“: Abnehmen können Sie uns, daß wir sechs verschiedene Disc-Programm-Pakete zusammengestellt haben, die von allererster Güte sind. Aber nun ist es doch an der Zeit, den „Vorhang zu lüften“ und die Programme – nebst Preisen – bekannt zu geben:

Bei diesem „KOMBI-DISC-ANGEBOT“ für den **SCHNEIDER CPC 464** handelt es sich um Spiele und Anwenderprogramme – alles nur vom feinsten! Autorennen / Computerschrift / Symbol Swap / Universal-Datei / Keyboard Toolkit / Farbdekodierung / Aladin / Painter / Box-Befehl und BASIC-Maschinen-Kit.
Dieses Paket kostet jetzt nur noch 56 Mark. Kaum zu glauben, was?

Bestell-Nr. US P1

Ja, das sind SUPER-GAMES, die ihrem Namen alle Ehre machen! Folgende MSX-Disketten sind enthalten: Zirkus / Pipeline / Alien Attack. Die Diskette gibt's für ganze 29,50 Mark! **Bestell-Nr. UM P1.** „Super Games“, die sich sehen lassen können, gibt es auch für den **ATARI:** Mauerflucht / Lunar-Simulation / Cherry Harry. Preis: „müde“ 24,50 Mark.

Bestell-Nr. UI P2

Beim „UTILITY SPECIAL“ kommen die **Commodore 64er-Freunde** voll auf ihre Kosten, die sich auf nur 35 „Steine“ belaufen. Was ist drin?
S-Tool 64 / Spritehilfe / Data-Generator / Diskmonitor. Wir meinen, ein tolles Angebot!

Bestell-Nr. UC P1

Die „BEST OF . . . DISCS“ sind ebenfalls für unsere **64er-Freunde** gedacht. Für 35 Mark erhalten Sie folgende „TIPS & TRICKS-BEST-SELLER“: Asmon / Re-Assembler / Maskengenerator / Multi BASIC V 3

Bestell-Nr. UC P2

„APPLE-UTILITY“, diese Bezeichnung sagt eigentlich schon alles. Was Sie aber zu einem wahren Super-Preis von 38,50 Mark geboten bekommen, wird Sie erstaunen! Dateiverwaltung / Disk / Cheque / DOS Change / Digital Sound Editor.

Bestell-Nr. UA P1

Die **„ATARI UTILITY“** ist ebenfalls Spitze, wir können dieses Disketten-Programm mit gutem Gewissen empfehlen! Und das zu einem Preis, den fast schon die Leer-Diskette kostet: 25 Mark! Zeichenersteller / Zeicheneditor / Monitorprogramm.

Bestell-Nr. UI P1

Da heißt es, nicht lange fackeln!

ZUGREIFEN!

Liebe Leser der „Tips u. Tricks von CPU“

Zunächst einmal möchte sich das Team von „Tips & TRICKS von CPU“ bei seinen Lesern und Programm-Autoren herzlich für das Interesse und die Unterstützung, die der Redaktion 1985 zuteil wurde, bedanken. Wir sind sicher, auch 1986 wieder aktuelle Informationen und „brauchbare“ Tips, Tricks und Programme zu „liefern“! („Siehe dieses Heft!“). In dieser Ausgabe beschäftigen wir uns – in gewohnter Manier – mit Utility-, Anwender- und Hilfsprogrammen. Unsere Highlights in dieser Ausgabe: es folgt der dritte Teil der Hardware-Serie „Universelles Bussystem / Bauanleitung; ferner starten wir den ersten Abschnitt eines „Maschinensprache-Kurses für den ZX-Spectrum; wir stellen das erste Anwenderprogramm für den Super-Rechner ATARI 520 ST („Mathematische Ableitungen“) vor; ein neues Prüfsummenverfahren (der „Checksummer“) wird helfen, den Fehlerteufel bei der Eingabe von Programmen endgültig zu verbannen (für Commodore 64); desweiteren testete unser Chef-Programmierer Frank Brall die derzeitig beliebteste Programmiersprache, TURBO PASCAL, für unsere Leser. Dies sind nur einige der Schwerpunkte der vorliegenden Ausgabe.

PS: Wir werden in Zukunft versuchen, die Computer der „neusten Generation“ besonders zu berücksichtigen.

Siegfried Görk, Chefredakteur

„Tips u. Tricks von CPU“ – Der Inhalt:

USER — Programme für den Anwender

		Seite
Dragon	3-D-Grafik	5
C-64:	Makro-Assembler	16
ZX-Spextrum:	Grafik-Compiler	25
C-64:	Speicherozilloskop	33
Apple:	RGS-Digital-Synthesizer	35
Atari:	Sektor-Kopierer	38

Utility — Hilfsprogramme für den Anwender

ZX-Spectrum:	Zeilenkill	43
Schneider CPC 464/664:	Datagenerator	43
Schneider CPC 464/664:	Toolkit	47
Turbo Pascal:	Test	49
Turbo Pascal:	ASCII-List	52
Atari:	Zeichensatz	53
Atari:	Translater	59
Atari 520 ST:	Mathematische Ableitungen	60
TI-99:	Basic-Erweiterungen	61
C-16/116	Merge-Routine	63
C-16/116	Old-Routine	65
C-64/VC-20/C-16/116:	Change Typ	62
VC-20:	Farb IRQ	66

HARDWARE

VC-20/C-64:	Universelles Bussystem/Bauanleitung (Teil 3)	68

„Tips und Tricks“-Extra

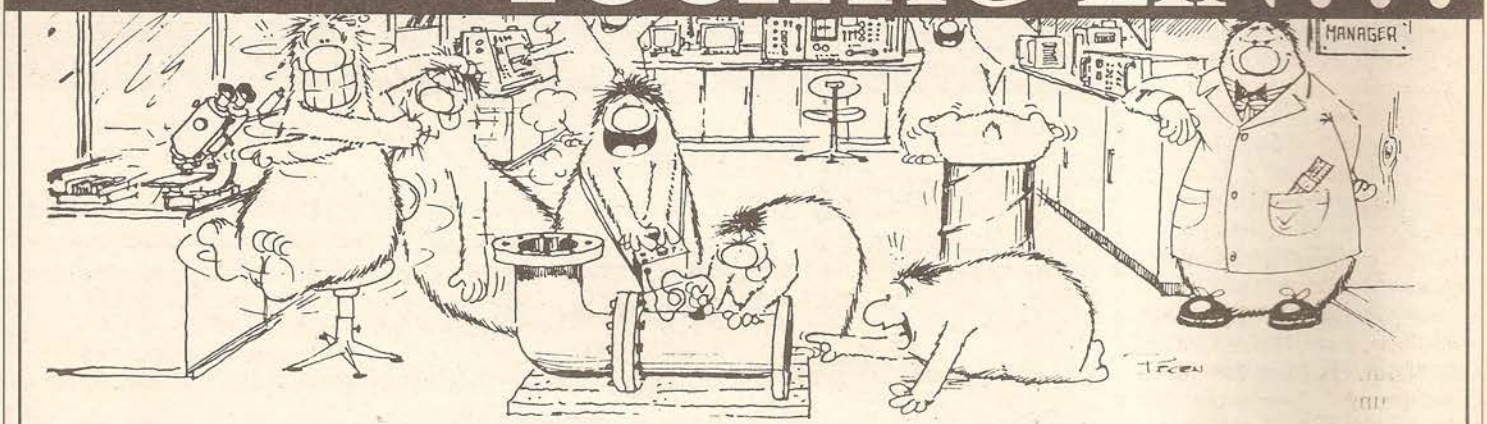
VC-20:	Drucker-Tips	76
C-64:	Scroller	77
TI-99:	Kleine Hilfen für den TI-99-Besitzer	78
C-64:	Goto X	37/79
SVI:	Schnellere Bildschirmausgabe	80

Rubriken

Computer-Club-Vorstellung BDVI-Vereinigung		86
Impressum		26
Infos	Atari 520 ST/Atari-News etc.	84
Software-Service		88/89
CPU-Basar		90

TRONIC-SONDERAUSGABE * TRONIC-SONDERAUSGABE
TOP-PROGRAMME * TOP-PROGRAMME

WIR „HEIZEN“ EUCH TÜCHTIG EIN...



Wie Sie sehen, sind die Redaktionen von HOMECOMPUTER und COMPUTRONIC schon kräftig bei der Vorbereitung zur tollen Winter-Sonderausgabe. Falls wir uns zuvor nicht „wiedersehen“: jetzt schon ein frohes Weihnachtsfest und ein glückliches 1986! (M.K.)

... mit der tollen SONDERAUSGABE der absoluten TOP-PROGRAMME aus HOMECOMPUTER ('85) und COMPUTRONIC ('84/'85)!

Ja, lieber Leser, wir bieten Ihnen eine Super-Spiele-Sonderausgabe, die es in sich hat: Genau die richtig „heißen“ Games zur naß-kalten Winterszeit. Auf über 150 Seiten werden wir a l l e TOP-PROGRAMME des Jahres 1985 – aus HOMECOMPUTER und COMPUTRONIC 1984/85 – veröffentlichen. Und das zu einem Preis, den man nicht ablehnen – ja, nicht widerstehen kann, präsentieren wir Spiele, Spiele, Spiele. Quasi „am laufenden Band“.

Doch aufgepaßt: Unsere SONDERAUSGABE hat nur eine begrenzte Auflage. Obwohl dieses einmalig günstige wie qualitative Heft von Dezember '85 bis Februar '86 im Handel sein wird, empfehlen wir jedem Heimcomputer-Fan sich schnell auf die Socken zu machen!

So werden wir Spiele, aller populären Heimcomputer-Systeme, der absoluten Spitzenklasse publizieren! Also, freuen Sie sich mit uns auf die „Winterausgabe“ der Superspiele vom TRONIC-Verlag. Wir sorgen dafür, daß Sie nicht in einen „Winterschlaf“ verfallen. So wie der Hamster beispielsweise.

Und: A propos Hamster – mit dieser SONDERAUSGABE haben Sie bestimmt genug „Vorrat“ für den ganzen Winter; dessen sind wir uns ganz sicher. (M.K.)

SUPER!

Grundlagen der 3D-Graphik in Maschinensprache am Beispiel des DRAGON 32

1. Perspektive

Die perspektivische Darstellung des Objekts ist einfach zu berechnen. In Abb. 1 ist der Bildschirm von der Seite zu sehen, das Innere rechts, das Auge des Benutzers links (Big Programmer's watching you ...). Die Z-Achse zeigt auf den Benutzer, die Y-Achse nach oben. Die X-Achse zeigt quasi auf uns und ist daher nicht zu sehen, d ist der Abstand des Benutzers zum Bildschirm, D der Abstand des Benutzers zum Koordinatenursprung, gemessen in Koordinateneinheiten.
Aus den dreidimensionalen Koordinaten x, y und z des Punktes P(x,y,z) wollen wir die zweidimensionalen Bildschirmkoordinaten x' und y' für den Bildpunkt P'(x',y') berechnen.
In Abb. 1 kann man sehen, daß gemäß dem Strahlensatz

$$\frac{y'}{d} = \frac{y}{D+z}$$ ist, also

$$y' = \frac{y \times d}{D+z}$$ ist, also

Analog ist

$$x' = \frac{x \times d}{D+z}$$

2. Rotation

Will man im zweidimensionalen Bereich ein Objekt drehen, benutzt man am besten Polarkoordinaten. Nach zahlreichen Versuchen stellte ich aber fest, daß die Übertragung ins Dreidimensionale nicht ohne weiteres möglich ist, da die Winkel nicht vollkommen unabhängig voneinander sind.
Um auf eine Alternatividee zu kommen, fertigte ich erst einmal eine Skizze an, in der ein Vektor A mit dem Winkel W um die Z-Achse gedreht wird, Abb. 2 (die Z-Achse zeigt auf uns und ist daher nicht zu sehen).
Was ich herausbekommen will, sind die x- und y- Koordinaten Ax' und Ay' des gedrehten Vektors A'. Aus den Hilfskonstruktionen in der Abbildung kann man erkennen:

Ax' = Ax x cos w - Ay x sin w,
Ay' = Ax x sin w + Ay x cos w.

Dies entspricht der Matrixmultiplikation

$$(Ax'\ Ay'\ Az') = (Ax\ Ay\ Az) * \begin{matrix} \cos w & \sin w & 0 \\ -\sin w & \cos w & 0 \\ 0 & 0 & 1 \end{matrix}$$

Diese Matrixmultiplikation beschreibt eine Drehung um die Z-Achse. Für Drehungen um die X- und Y-Achse gelten die folgenden Matrizen:

$$\begin{matrix} 1 & 0 & 0 \\ 0 & \cos w & \sin w \\ 0 & -\sin w & \cos w \end{matrix} \quad \text{und} \quad \begin{matrix} \cos w & 0 & \sin w \\ 0 & 1 & 0 \\ -\sin w & 0 & \cos w \end{matrix}$$

$$(Ax'\ Ay') = (Ax\ Ay) * \begin{matrix} \cos w & \sin w \\ -\sin w & \cos w \end{matrix} .$$

(Ein paar Sätze zu Matrixmultiplikationen:
Die Zeilen der ersten Matrix werden Zeile für Zeile mit allen Spalten der zweiten Matrix multipliziert, und zwar jeweils das erste Element der momentanen Zeile mit dem ersten Element der momentanen Spalte, das zweite Element der Zeile mit dem zweiten der Spalte usw.. Die Summe der Elementprodukte wird in die dritte Matrix geschrieben. Das hört sich kompliziert an, ist es aber nicht.

Ein Beispiel:

$$\begin{matrix} a & b & c \\ d & e & f \\ g & h & i \end{matrix} * \begin{matrix} j & k & l \\ m & n & o \\ p & q & r \end{matrix} = \begin{matrix} aj+bm+cP & ak+bn+cq & al+bo+cr \\ dj+em+fP & dk+en+fq & dl+eo+fr \\ gj+hm+iP & gk+hn+iq & gl+ho+ir \end{matrix}$$

Wenn ich die Rotation von Vektor A in den Vektor A' in den dreidimensionalen Raum übertragen will, muß ich die Matrizen ebenfalls (mindestens) dreizeilig und dreispaltig gestalten (die homogenen Koordinaten mit vier Zeilen sind zwar für weitere Transformationen praktisch, für die Rotation reichen aber dreizeilige).

Viele haben den Dragon schon abgeschrieben. Doch „Totgesagte" leben bekanntlich länger. Was der Dragon-Computer zu bieten hat, können Sie unschwer am Programm: „3-D-Graphik" erkennen.

3. M-Code: Variablen

Die Variablen liegen ab der Speicheradresse 22000 dezimal. Mit DOTEND wird von BASIC aus die Endadresse der ebenfalls von BASIC aus angelegten Punktetabelle angegeben, die bei DOTBEG beginnt.
SIN und COS sind Sinus und Kosinus des Drehwinkels und werden von BASIC aus in folgendem Format gepokt:

Vorzeichenbit, Vorkommabit, 14 Nachkommabits.

Der Wert 1 hat nach diesem Format die hexadezimale Darstellung $4000, der Wert -0.5 $A000. Diese Darstellung hat den Vorteil, daß die möglichen Sinus- und Kosinuswerte (-1 bis 1) so genau wie möglich in 16 Bits gespeichert werden können.

SCRBEG und SCREND geben die Anfangs- und Endadresse des Bildschirmspeichers an, auf dem momentan gezeichnet wird. Da nach Vollendung einer Figur dieser Bildschirm sichtbar gemacht wird und auf einem zweiten die nächste Drehung vorbereitet wird, geben SCRBEG und SCREND normalerweise gerade den nicht sichtbaren Graphikbildschirm an.

MFLAG, F1, F2 und PROD sind Zwischenspeicher für Multiplikationen. DOTPTR zeigt auf

den gerade behandelten Punkt der Punktetabelle. XYPTR zeigt auf den gerade errechneten Punkt der Tabelle mit den zweidimensionalen Bildschirmkoordinaten der Punkte, die bei PUNBEG hinter dem Programm beginnt.
XMEM, YMEM und ZMEM sind Zwischenspeicher für einen Punkt bei diversen Berechnungen.
LMNPTR zeigt auf die zu benutzende Rotationsmatrix, und zwar M1LMN für die Rotation um die Z-, M2LMN um die Y- und M3LMN um die X-Achse, NORMMT für keine Rotation. Die Matrizen sind vom Assembler noch mit dem Drehwinkel 0 Grad definiert. Sie werden bei INIT mit den entsprechenden Sinus- und Kosinuswerten versehen. NRMEND gibt nur das Ende der Normalmatrix an.
ZWMAT ist eine Hilfsmatrix für Matrixoperationen.
MABC ist die Matrix eines mit der Figur mitdrehenden Koordinatensystems. Damit nicht jeder Punkt der Figur mit 16 Bit Genauigkeit gedreht werden muß (dauert zu lange), wird das 16-Bit-Koordinatensystem KABC, das durch die Matrix MABC festgelegt wird, mit der Figur mitgedreht. Um die Koordinaten der gedrehten Punkte zu erhalten, muß man ihre ursprünglichen Koordinaten nur mit den Elementen aus MABC multiplizieren, als ob MABC eine normale Rotationsmatrix wäre. MABC legt die Einheitsvektoren A, B und C (Koordinatenachsen der Länge 1) so fest:

$$MABC = \begin{matrix} Ax & Ay & Az \\ Bx & By & Bz \\ Cx & Cy & Cz \end{matrix}$$

Die Berechnung des gedrehten Punktes P'(x',y',z') aus den Koordinaten des ursprünglichen Punktes P(x,y,z) geschieht folgendermaßen:

$$P' = P \times MABC \text{ , also}$$

$$(x'\ y'\ z') = (x\ y\ z) \times \begin{matrix} Ax & Ay & Az \\ Bx & By & Bz \\ Cx & Cy & Cz \end{matrix}$$

KEYMEM enthält den ASCII-Code der zuletzt gedrückten Taste. Bei DOTBEG beginnt, die von BASIC aus gepokete Punktetabelle. In ihr wird jeder Punkt durch vier Bytes festgelegt: die drei 8-Bit-Koordinaten (7 Bit mit Vorzeichen) und ein Lineflag. Dieses Lineflag ist 1 beim ersten Punkt eines Linienzuges, 0 bei jedem weiteren Punkt im Linienzug und 3 beim letzten Punkt der ganzen Figur. Beispiel:
Stehen in der Tabelle die Werte 0,0,0,1, 0,0,1,0, 0,0,4,1, 0,3,4,3, besteht die Figur aus zwei Strecken.

4. Hauptprogramm

Das Haupt- oder Rahmenprogramm des Maschinenspracheprogramms beginnt bei 27000 dezimal.
Zuerst wird die Initialisierungsroutine aufgerufen. Dann wird bei LOOP mit der Tastaturroutine eine Taste abgefragt. Wenn BREAK gedrückt wurde (mit dem Code 3), wird zu BASIC zurückgekehrt.
Ansonsten wird bei WEITER geprüft, ob die Taste „R" („Restart") gedrückt wurde. Wenn ja, wird das komplette Maschinenspracheprogramm von vorne gestartet; die Figur erscheint in der Ausgangslage.
War das nicht der Fall, wird mit der Routine WAHL die gewünschte Rotationsmatrix gewählt, mit DREH das mitdrehende Koordinatensystem MABC gedreht, mit POSI die Tabelle mit den Bildschirmkoordinaten errechnet, mit DRAW die Linienzüge durch die Punkte gezeichnet, mit SWITCH der Bildschirm umgeschaltet und wieder zu LOOP zurückgesprungen.

5. Initialisierung

Zuerst wird bei INIT die Normalmatrix in die mitdrehende Matrix MABC kopiert, diese also in die Ausgangslage gesetzt. Dann wird der erste Graphikbildschirm ausgewählt und mit Hilfe der Softswitches 65472 bis 65479 sichtbar gemacht. Anschließend wird mit der Routine PCLS der gewählte Bildschirm gelöscht.
Schließlich werden die von BASIC übergebenen Werte für den Sinus und den Kosinus des Drehwinkels in die Rotationsmatrizen eingesetzt.

6. Tastaturabfrage

Hier wird das einzige Mal eine ROM-Routine benutzt. Bei $8006 wird der ASCII-Code der gerade gedrückten Taste ermittelt. War eine Taste gedrückt, wird ihr Code nun in KEYMEM gespeichert.

7. Wahl der Rotationsmatrix

Bei WAHL wird das X-Register erst einmal mit der Adresse der Normalmatrix geladen. War eine der Tasten 1, 2 oder 3 zuletzt gedrückt, wird die Adresse der entsprechenden Rotationsmatrix in das X-Register geladen. Dieses wird schließlich im LMNPTR abgespeichert.

8. Drehung der Matrix MABC

Bei DREH werden zuerst das X-, Y- und U- Register mit den Adressen der Matrix MABC, der gewählten Rotationsmatrix und der Hilfsmatrix ZWMAT (für das Ergebnis) geladen. Das Unterprogramm MATRIX führt die Matrixmultiplikation aus. Anschließend wird das Ergebnis aus ZWMAT wieder in MABC hineinkopiert.
Schließlich werden die Tabellenzeiger DOTPTR und XYPTR auf den Anfang ihrer Tabellen gesetzt.
Die Routine MATRIX multipliziert zwei dreireihige Matrizen miteinander, deren Adressen im X- und im Y- Register stehen. Das Ergebnis wird ab der Adresse gespeichert, die im U-Register steht.
Die Routine besteht aus drei ineinandergeschachtelten Schleifen, in deren Mitte die Multiplikation zweier Elemente mit der Routine MULS und die anschließende Addition zu den bisherigen Elementprodukten derselben Reihe steht. Da wir aber ein spezielles Format benutzen (1 Bit Vorzeichen, 1 Bit vor dem Komma, 14 Bit nach dem Komma), muß das Ergebnis der Multiplikationsroutine MULS im D- Register und in PROD vor der Addition um zwei Stellen nach links verschoben werden.
Da ich die Routine MATRIX einmal eventuell für ein verschiebbares Programm benutzen wollte, sind die Zwischenspeicher auf dem Stack eingerichtet. Deshalb die beiden LEAS- Instruktionen.
Die Routine MULS multipliziert zwei 16-Bit-Zahlen mit Vorzeichen in F1 und F2 und liefert ein 32-Bit-Ergebnis mit Vorzeichen in PROD bis PROD+3. Wer mehr Geschwindigkeit will, kann die Berechnung der niedrigsten Bits auch weglassen.

8. Bildschirmkoordinaten

Bei POSI werden die Punktkoordinaten aus der Original-Punktetabelle herausgelesen (mit dem X-Register als Zeiger), durch Matrixmultiplikation mit MABC durch die Unterroutine MINMAT gedreht, in XMEM, YMEM und ZMEM zwischengespeichert und in die Routine PERSPEK gegeben, die die zweidimensionalen Bildschirmkoordinaten errechnet. Diese werden in die PUNTAB-Tabelle gespeichert (mit XYPTR als Zeiger).
MINMAT multipliziert die 8-Bit-3*3-Matrix, auf die das X-Register zeigt, mit der 16-Bit-3*3-Matrix, auf die das Y-Register zeigt. Das Ergebnis wird ab der durch das U- Register festgelegten Adresse gespeichert. In der Routine MINMAT werden wieder Variablen auf dem Stack gespeichert.
Die Routine MINMAT besteht aus zwei ineinandergeschachtelten Schleifen, in deren Zentrum, wie bei MATRIX, die Ad-

dition der Produkte je zweier Elemente steht, diesmal durch die 8*16-Bit-Multiplikationsroutine MINMUL, deren 16-Bit-Ergebnis in MPROD steht.
Bei PERSPEK werden X und Y durch das D- Register übergeben: Z + 128 muß vor dem Aufruf auf den Stack gepusht worden sein. Die Variablen werden wiederum auf dem Stack gespeichert.
Gleich zu Anfang wird der Nenner D + z errechnet. Dabei wird D mit $480 festgelegt und zu z addiert, indem Z+128 ins B- und 4 ins A- Register geladen wird. D wird als Divisor gespeichert, dann wird X Positiv gemacht und ins Highbyte des Dividenden geladen, also mit $100 multipliziert. Da wir aber d = $400 setzen wollen, wird zwar nicht der Dividend mal 4 genommen, aber der Divisor ins D- Register geladen und um zwei Stellen nach rechts geschoben. Nach dem Aufruf der 16/16-Bit-Division DIV wird das 8-Bit-Ergebnis in B nötigenfalls negativ gesetzt, zur Bildschirmmitte (128) addiert und als XBILdpunkt gespeichert. Analog wird mit Y verfahren. Schließlich werden XBild und YBild ins D- Register geladen.
Die Routine DIV verlangt bei Aufruf den Dividenden auf dem Stack und den Divisor in D (beide 16 Bit). Das Ergebnis in D ist eigentlich 16 Bit groß, aber da bei unserer Anwendung der Nenner nicht kleiner als $100 ist, paßt es immer in die 8 Bit des B- Registers.

9. Zeichnen der Linien

Bei DRAW wird zuerst das X-Register auf den Anfang der Tabelle mit den zweidimensionalen Bildschirmkoordinaten gesetzt. Dann werden die X- und die Y- Koordinate eines Punktes ins D- Register geladen. Anschließend wird das Linienflag dahinter getestet. War es gleich 0, wird bei DRAW1 eine Linie vom letzten Punkt zu dem momentanen gezogen. War es nicht gleich 0, werden vorher die Koordinaten des letzten Punktes in $BE/$C0 durch die Koordinaten des momentanen Punktes ersetzt; d. h., es wird eine „Linie" von diesem Punkt zu sich selbst gezogen: ein neuer Linienzug wird begonnen.
Anschließend wird das Linienflag noch mit 3 verglichen. Ist es gleich 3, war dies der letzte Punkt, und es wird zum Hauptprogramm zurückgesprungen, andernfalls wird bei DRAWLP der nächste Punkt vorgenommen.
Die LINE-Routine basiert auf dem Bresenham-Algorithmus. Das Prinzip dabei ist folgendes: Wenn von einem Punkt zum anderen ein Schritt in X-Richtung gemacht wird, wird zu einem Mittelwert (im Programm OFFSET) DY addiert. Wird OFFSET zu groß, wird ein Schritt in Y-Richtung gemacht und DX subtrahiert. Dabei kommt eine Treppe heraus, die der gewünschten Linie sehr nahekommt, ohne daß man Multiplikation oder Division braucht. Das Verfahren beruht darauf, daß DX Schritte in X-Richtung multipliziert mit DY dasselbe ergeben wie DY Schritte in Y-Richtung multipliziert mit DX.
Bei LINE werden zuerst die Variablen XALTER und YALTER (auf dem Stack) gelöscht. Dann werden die neuen Koordinaten aus dem D-Register in XNEU und YNEU gespeichert. Die alten Koordinaten werden aus $BE und $CO geholt und zu der Bildschirmadresse im X-Register und der Bitmaske in MASKE umgerechnet. Bei PSPOKE wird der alte Punkt auf dem Bildschirm sichtbar gemacht.
Bei LINO wird getestet, ob XNEU-XALT positiv ist. Wenn ja, wird XINC (Schrittweite in X-Richtung) auf 1 gesetzt, wenn nein, auf -1. In beiden Fällen wird DX = ABS (XNEU - XALT) gesetzt. Bei LIN2 geschieht ähnliches in Y-Richtung, allerdings mit der Schrittweite 8/- 32, was dem Adressenabstand einer Bildschirmzeile entspricht.

Bei LIN4 wird geprüft, ob DX größer als DY ist. Wenn nein, werden DX und DY vertauscht, XINC und YINC gelöscht und durch die alternativen Schrittweiten YALTER und XALTER ersetzt. Dadurch wird später automatisch in Y-Richtung geschritten und gegebenenfalls ein Schritt in X-Richtung durchgeführt.
Bei LIN5 wird der Mittelwert OFFSET auf DX/2 gesetzt und der Schrittzähler LINCTR auf 1.
Bei LINLOP wird XINC getestet. Ist es größer als 0, erfolgt ein Schritt nach rechts (die Maske um ein Bit verschoben und ggf. in die nächste Bildschirmadresse gewechselt), ist es kleiner als 0, nach links. Ist es gleich 0, ist XALTER gleich +/- 32, und bei LIN8 erfolgt ein Schritt in Y-Richtung.
Dann wird geprüft, ob LINCTR zu hoch geworden ist. Das ist ein „Notausstieg", der noch vom Testen stammt, der aber auch Fehler abfangen kann, wenn unzulässige Werte an die LINE-Routine übergegen werden (bei anderen Anwendungen). Anschließend wird DY zum Offset addiert (s. o.) und, wenn dies zu groß geworden ist, erfolgt bei LIN7 ein Schritt in die Richtung der kleineren Differenz (oben DY). War diese DX, also in X-Richtung, wird die Maske nach rechts oder links geschoben und ggf. in das nächste Bildschirmbyte gewechselt, war sie DY, wird durch die Addition von YINC zur Adresse eine Bildschirmzeile nach oben oder unten gegangen.
Bei LIN6 wird zuerst sicherheitshalber geprüft, ob die Adresse innerhalb des Bildschirmbereiches liegt, und wenn ja, wird der momentane Linienpunkt durch OR-Verknüpfung der Maske mit dem Bildschirmbyte sichtbar gemacht.
Bei LIN12 wird gesehen, ob schon DX Punkte gezeichnet worden sind. Wenn ja, wird der Stack restauriert und die neuen Koordinaten als alte in $BE und $CO abgespeichert.

10. Bildschirmumschaltung

Bei SWITCH wird die momentane Bildschirmanfangsadresse SCRBEG mit der Anfangsadresse des Schirm 1 verglichen. Stimmt sie überein, wird Schirm 1 sichtbar gemacht und die Adressen von Schirm 2 in SCRBEG und SCREND gespeichert, um dort weiterzuzeichnen, andernfalls wird Schirm 2 sichtbar gemacht und Schirm 1 zum Weiterzeichnen vorbereitet.

11. BASIC-Ladeprogramm

Wer bis hierher in etwa folgen konnte, hat bestimmt keine Probleme, den BASIC-Lader zu analysieren. Deshalb nur ein paar kurze Sätze:
In den Zeilen 20, 30 und 160-210 findet die Initialisierung statt. FNH(X) und FNL(X) teilen die Zahl X in High- und Lowbyte auf. ME ist die Adresse von SPEIBG = DOTEND, DO die Anfangsadresse der Punktetabelle.
Zeile 40-150 zeigt das Menü.
Zeile 1000-1050 übergibt den Drehwinkel an das Maschinenspracheprogramm und ruft es auf.
In den Zeilen 2000-2270 befindet sich ein 3D-Editor, bei dem man abwechselnd den Grund- und den Seitenriß sieht. Im Grundriß kann man mit den Cursortasten (mit SHIFT in Zehnerschritten) die X- und die Y-Koordinate festlegen. Dann drückt man „S", wenn ein neuer Linienzug begonnen werden soll, „L", wenn ein alter Linienzug weitergeführt werden soll, „CLEAR" für das Löschen des letzten Punktes (ist im Editor leider nicht zu sehen; wäre zu kompliziert für den Nutzen) und „E" für den Abschluß des Editierens. Im Seitenriß legt man auf die gleiche Weise die Z-Koordinate fest und drückt die gleiche Taste („S" oder „L") noch einmal.
Wurde „E" gedrückt, wird in Zeile 2110 der letzte Punkt verdoppelt und mit dem Ende-Lineflag 3 versehen, ferner wird die Endadresse der Tabelle nach DOTEND bzw. ME gepokt.
In Zeile 3000-3080 wird eine Fi-

gur als „Maschinenspracheprogramm“ auf Kassette gespeichert, in Zeile 4000-4050 von Kassette geladen.
In Zeile 5000-5140 kann eine eigene Hardcopyroutine aufgerufen werden, die ab Adresse 20000 im Speicher stehen muß. Vorher werden abwechselnd beide Bildschirme gezeigt, bis „H“ gedrückt wird, damit der richtige Bildschirm ausgedruckt wird. Dann wird ein Kasten um das Bild gezeichnet (nicht bis ganz unten, da meine Hardcopyroutine die letzten drei Zeilen abschneidet) und nach weiterem Tastendruck der Drucker initialisiert und die Hardcopyroutine aufgerufen.
In Zeile 6000 wird der Bildschirm gelöscht und das Programm beendet.

12. Eingabe des Programms

1. ROTATE-HEXLOADER eintippen und sicherheitshalber abspeichern.
2. eingeben: RUN
3. 3D-ROTATE-HEXDUMPS eingeben. Wenn etwas nicht stimmt, wird die Zeile nochmals abgefragt. Zum Schluß wird das Maschinenspracheprogramm auf Kassette gespeichert.
4. BASIC-LADER eintippen und auf Kassette speichern.

Wenn alles eingegeben und abgespeichert ist, kann man das Programm so starten:
1. eingeben: CLEAR 200,19999
2. mit CLOADM“ROTATE“ das Maschinenprogramm laden.
3. mit CLOAD“ROTBASIC“ das BASIC-Programm laden.
4. eingeben: RUN

Die Funktion des Editors ist unter 10. schon beschrieben, der Rest des BASIC-Programms erklärt sich von selbst. Hier noch einmal die Funktionen des Maschinenprogramms:
1,2,3 : Drehen um drei Achsen,
R : Ausgangslage,
BREAK : Ende.
Der Editor scheint zuerst etwas kompliziert zu sein, man gewöhnt sich aber schnell daran. Deshalb ein Beispiel: ein Würfel soll konstruiert werden. (Im folgenden bedeuten die Buchstaben o, u, r und l, daß die Pfeiltaste nach oben, unten, rechts bzw. links zusammen mit Shift (!) so oft gedrückt werden, wie die Zahl davor angibt.) Nach der Eingabe von RUN werden folgende Tasten gedrückt (die Kommentare in Klammern natürlich nicht!):
N 8r8u (Startposition in einer Ecke)
16oLL 16rLL 16uLL 16lLL (Bodenfläche)
L16oL (Seitenkante zur Dachfläche)
16oLL 16rLL 16uLL 16lLL (Dachfläche)
16oSS L16uL 16rSS L16oL 16uSS L16uL (die drei restlichen Seitenkanten, abwechselnd von oben nach unten und umgekehrt gezeichnet)
E (Ende des Editierns)
D (Drehen)
10 <Enter> (Drehwinkel)
1 2 3 R 1 2 3 ... (Drehen des Würfels nach Belieben)

13. Literatur

Myers, Roy E., Microcomputer Graphics, ist edition, 4th Printing, 1982, Addison Wesley Publishing Company Inc., Massachusetts (US), ISBN 0-201-05092-7,
behandelt Transformationen zwei- und dreidimensionaler Objekte, Clipping, Perspektive und einen eingeschränkten Algorithmus für Hidden-Line-Elimination; mit Beispielen für den Apple II.
Newman, William M. / Sproull, Robert F., Principles of Interactive Computer Graphics, 2nd edition, 9th Printing, 1979, McGraw-Hill Book Company Japan, ISBN 0-07-046338-7,
behandelt verschiedene Display-Hardware, Linien-zeichen-Algorithmen, Clipping, Fenstertechnik, Kreisalgorithmen, zwei- und dreidimensio-

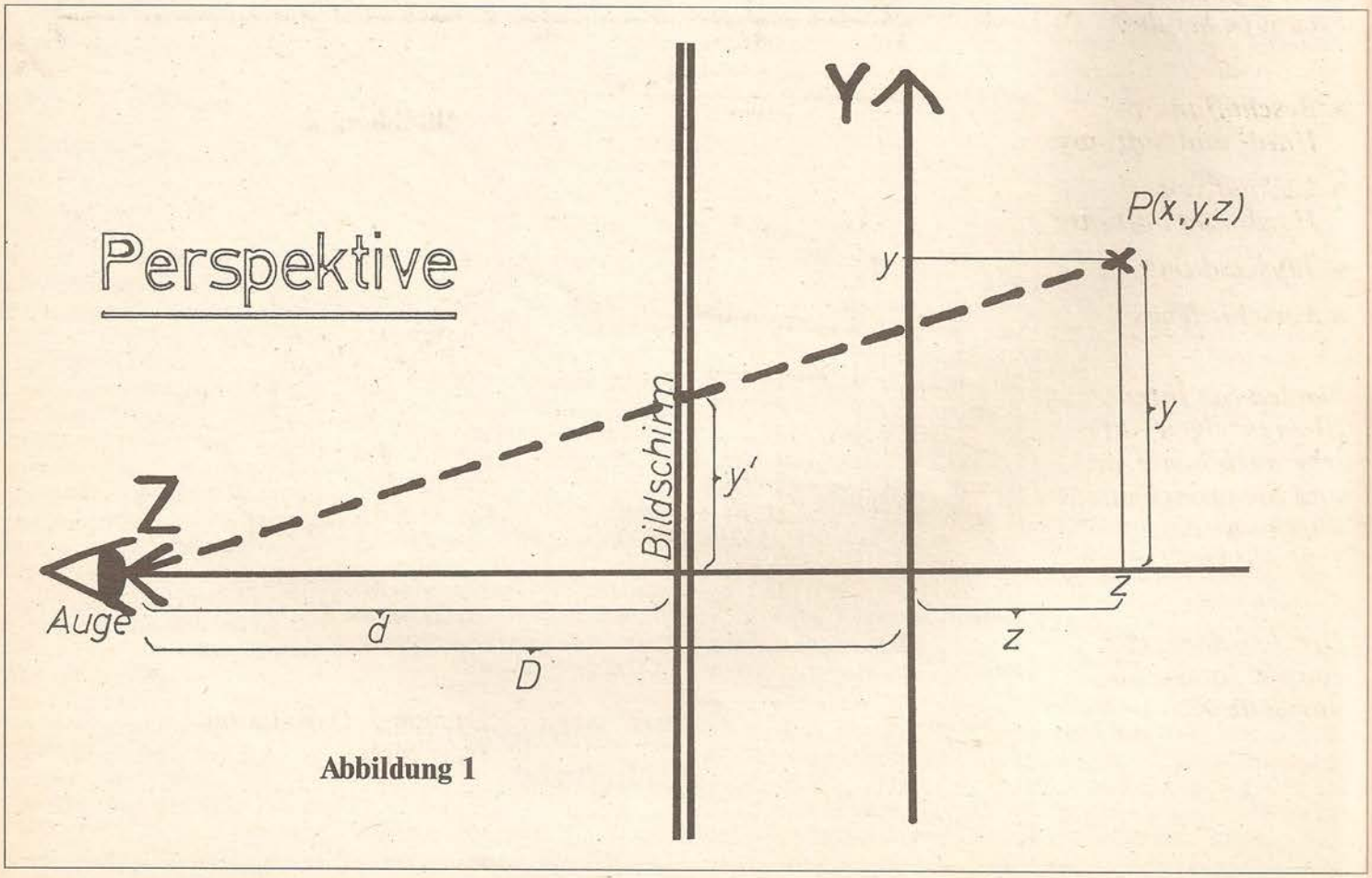


Abbildung 1

nale Transformationen, Polygon-Clipping, Segmenttechnik, Desplay-File-Aufbau, Entwurf eines Graphik-Pakets für ein Betriebssystem o. ä., interaktive Graphik, Rastergraphik, Perspektive, Hiddenline- und Hidden-Surface-Elemination, Schattierung und vieles mehr. Dieses Buch ist sehr grundlegend geschrieben und wohl eher für Informatikstudenten o. ä. geeignet.
Beide Titel gibt es inzwischen auch in Deutschland. Beide haben einen Anhang, in dem Matrixrechnung und Vektorrechnung näher erläutert werden.

Kleinanzeigen lösen Probleme!

Für nur 10,– DM pro Anzeige hilft „CPU" allen Computer-Freunden bei der

- ***Beschaffung von Hard- und Software***
- ***Verkauf von Hard- und Software***
- ***Tauschaktion***
- ***Verschiedenes***

Senden Sie Ihren Kleinanzeigen-Auftrag noch heute ein und Sie erreichen durch uns eine Vielzahl von Lesern.

Den Kleinanzeigencoupon finden Sie auf Seite 42!

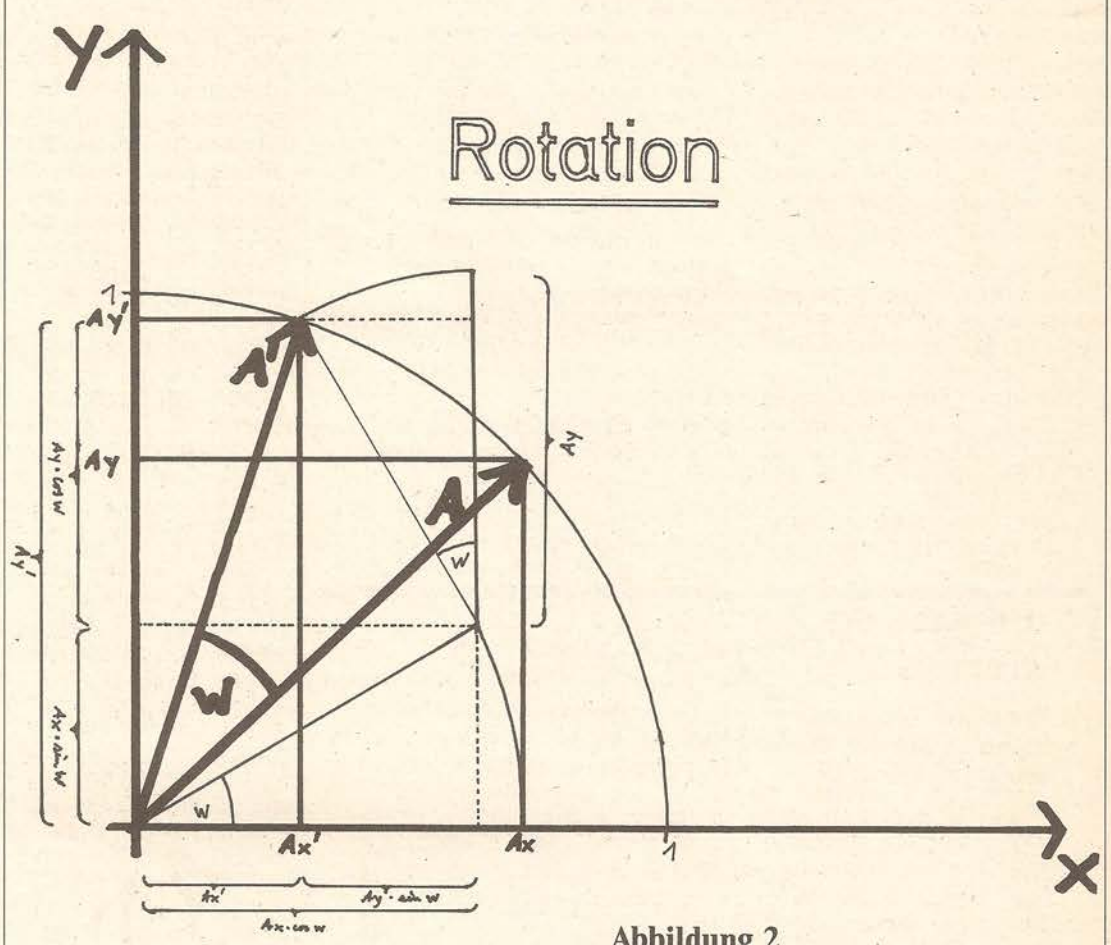
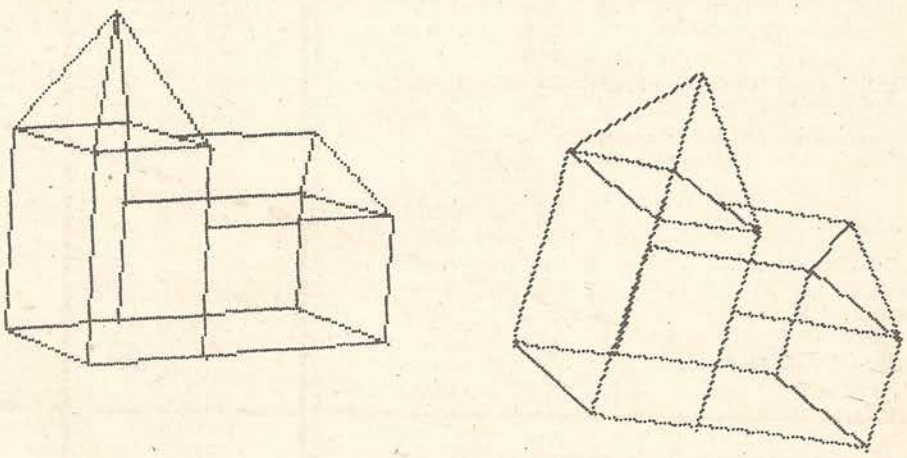


Abbildung 2



Dragon-3-D-Rotation: Hardcopies

```
10 '****************************
20 '**** ROTATE- HEXLOADER ****
30 '****************************
40 '
50 CLEAR 200,21999:CLS
55 ST=22000:EN=22340
60 FOR I=ST TO EN STEP 10
70 C=0
80 PRINT"zeile";I
90 FOR J=0 TO 9
100 INPUT "HEX-WERT";H$
110 P=VAL("&H"+H$):POKE I+J,P:C=C+P
120 NEXT J
130 SOUND 200,3:INPUT"PRUEFSUMME";O
140 IF O<>C THEN PRINT"FEHLER!":PRINT" NOCHMAL";:P
LAY"T1603CE-F#A":GOTO 70
150 CLS:NEXT I
200 PLAY"T1601CEGO5CEG":CLS
210 IF ST<27000 THEN ST=27000:EN=28070:GOTO 60
220 PRINT@32*4," CASSETTENRECORDER AUF AUFNAHME  S
TELLEN, DANN TASTE DRUECKEN!"
230 AUDIOON
240 IF INKEY$="" THEN 240
250 CSAVEM"ROTATE",22000,28080,27000
```

3D-ROTATE-Hexdump

mit Prüfsummen

```
22000 57 3C 6 B1 3F A6 1E 0 35 FF = 897
22010 0 2A C6 3F A6 A A2 D8 64 56 = 1043
22020 B8 6E 8 3D BC A CC 56 31 3F = 963
22030 A6 6 B1 0 0 F9 4F 3F A6 0 = 906
22040 0 0 0 0 0 40 0 3F A6 0 = 293
22050 0 6 B1 0 0 40 0 0 0 F9 = 496
22060 4F 0 0 3F A6 40 0 0 0 0 = 372
22070 0 0 0 3F A6 6 B1 0 0 F9 = 661
22080 4F 3F A6 40 0 0 0 0 0 0 = 372
22090 0 25 94 33 BF 0 0 CC 35 25 = 721
22100 8F 40 0 0 0 0 0 0 0 25 = 244
22110 94 33 BF 0 0 CC 35 25 8F 40 = 891
22120 0 0 0 0 0 0 0 40 0 0 = 64
22130 0 0 0 0 0 40 0 3 0 0 = 67
22140 0 0 0 0 0 0 0 0 0 0 = 0
22150 0 0 FF FF FF FF FF DF FF FF = 2008
22160 FF FF FF FF FF FF FF FF FF FF = 2550
22170 FF FF FF FF FF FF FF FF FF FF = 2550
22180 FF FF FF FF FF FF FF FF FF FF = 2550
22190 FF FF FF FF FF FF FF FF FF FF = 2550
22200 C2 E1 3D 1 3D E1 3D 0 3D 1E = 919
22210 3D 0 C2 1E 3D 0 C2 E1 3D 0 = 826
22220 C2 E1 F0 0 F0 E1 F0 0 F0 1E = 1634
22230 F0 0 C2 1E F0 0 C2 E1 F0 0 = 1363
22240 D9 0 C2 0 C2 1E F0 0 F0 1E = 1145
22250 F0 1 D9 0 C2 0 F0 E1 F0 0 = 1357
22260 F0 E1 3D 0 F0 1E 3D 1 F0 1E = 1128
22270 F0 0 C2 1E F0 1 C2 1E 3D 0 = 990
22280 3D 1E 3D 1 3D 1E F 0 F0 1E = 529
22290 F 0 F0 E1 F 1 3D E1 F 0 = 797
22300 3D 1E F 0 2E 0 F0 0 3D E1 = 678
22310 F 0 2E 0 F0 1 F0 0 F0 0 = 782
22320 3D E1 F 1 3D E1 3D 0 3D E1 = 935
22330 3D 3 FE FE FE FF FF FF FF FF = 2101
22340 FF 20 4C 49 4E 31 30 57 38 10 = 770


27000 17 0 1D 17 0 89 81 3 26 1 = 383
27010 39 81 52 27 F1 17 0 86 17 0 = 728
27020 A2 17 1 9C 17 2 C6 17 3 E0 = 815
27030 20 E3 8E 56 67 10 8E 56 55 EC = 1155
27040 81 ED A1 8C 56 79 25 F7 CC 6 = 1368
27050 0 FD 55 F6 CC 1D FF FD 55 F8 = 1658
27060 4F B7 FF C0 B7 FF C3 B7 FF C5 = 1977
27070 B7 FF C7 86 F7 B7 FF 22 17 0 = 1513
27080 30 FC 55 F2 FD 56 F FD 56 23 = 1355
27090 FD 56 3B 43 50 25 1 4C FD 56 = 998
27100 13 FD 56 2B FD 56 3F FC 55 F4 = 1384
27110 FD 56 D FD 56 1F FD 56 39 FD = 1371
27120 56 15 FD 56 2F FD 56 41 39 BE = 1144
27130 55 F6 CC 0 0 ED 81 BC 55 F8 = 1422
27140 25 F9 39 BD 80 6 27 3 B7 56 = 977
27150 79 39 B6 56 79 8E 56 67 81 31 = 1076
27160 26 3 8E 56 D 81 32 26 3 8E = 644
27170 56 1F 81 33 26 3 8E 56 31 BF = 806
27180 56 B 39 8E 56 55 10 BE 56 B = 770
27190 CE 56 43 17 0 1D 8E 56 43 10 = 722
27200 8E 56 55 EC 81 ED A1 8C 56 55 = 1387
27210 25 F7 CC 56 B8 FD 56 3 CC 6D = 1413
27220 A5 FD 56 5 39 34 76 32 7D 86 = 1045
27230 3 A7 E4 86 3 A7 61 86 3 A7 = 1103
27240 62 6F C4 6F 41 EC 81 FD 55 FB = 1535
27250 EC A4 31 26 FD 55 FD 17 0 2B = 1144
27260 FC 55 FF 78 56 1 59 49 78 56 = 1167
27270 1 59 49 E3 C4 ED C4 6A 62 26 = 1261
27280 DC 31 30 30 1A 33 42 6A 61 26 = 749
27290 CA 31 3A 30 6 6A E4 26 BE 32 = 975
27300 63 35 F6 7F 55 FA 7D 55 FB 2C = 1365
27310 E 73 55 FA 73 55 FB 70 55 FC = 1364
27320 25 3 7C 55 FB 7D 55 FD 2C E = 1021
27330 73 55 FA 73 55 FD 70 55 FE 25 = 1391
27340 3 7C 55 FD 7F 55 FF 7F 56 0 = 1145
27350 B6 55 FC F6 55 FE 3D FD 56 1 = 1505
27360 B6 55 FB F6 55 FE 3D F3 56 0 = 1493
27370 FD 56 0 B6 55 FC F6 55 FD 3D = 1503
27380 F3 56 0 FD 56 0 B6 55 FB F6 = 1432
27390 55 FD 3D F3 55 FF FD 55 FF 7D = 1700
27400 55 FA 27 1F 73 55 FF 73 56 0 = 1061
27410 73 56 1 70 56 2 25 11 FC 56 = 794
27420 0 C3 0 1 FD 56 0 B6 55 FF = 1057
27430 89 0 B7 55 FF 39 8E 56 B8 10 = 1145
27440 8E 56 55 CE 56 7 17 0 25 B6 = 854
27450 56 9 8B 80 34 2 FC 56 7 17 = 784
27460 0 9E 10 BE 56 5 ED A1 35 2 = 908
27470 A6 3 A7 A0 10 BF 56 5 30 4 = 846
27480 BC 55 F0 25 D2 39 34 76 32 7D = 1162
27490 86 3 A7 61 86 3 A7 62 6F C4 = 1110
27500 6F 41 A6 80 B7 6B DF EC A4 31 = 1432
27510 26 FD 6B E0 17 0 1D FC 6B E2 = 1259
27520 58 49 58 49 E3 C4 ED C4 6A 62 = 1382
27530 26 E2 31 30 30 1D 33 41 6A 61 = 757
27540 26 D0 32 63 35 F6 7F 6B DE 7D = 1275
27550 6B DF 2C 6 73 6B DE 70 6B DF = 1266
27560 7D 6B E0 2C E 73 6B DE 73 6B = 1180
27570 E0 70 6B E1 25 3 7C 6B E0 78 = 1283
27580 6B E1 79 6B E0 B6 6B E0 F6 6B = 1650
27590 DF 3D 44 56 FD 6B E2 7D 6B DE = 1478
27600 27 B 73 6B E2 70 6B E3 25 3 = 984
27610 7C 6B E2 39 0 3D 4B 1E 8 EF = 927
27620 32 77 ED 66 86 4 E6 6B ED 62 = 1318
27630 6F 68 A6 66 2E 3 40 63 68 5F = 894
27640 ED E4 EC 62 47 56 47 56 17 0 = 1136
27650 2C 6D 68 27 1 50 CB 80 E7 64 = 1039
27660 6F 68 A6 67 2E 3 40 63 68 5F = 895
27670 ED E4 EC 62 47 56 47 56 17 0 = 1136
27680 E 6D 68 27 1 50 CB 80 E7 65 = 1010
27690 EC 64 32 69 39 34 6 86 10 34 = 808
27700 2 4F 5F 68 66 69 65 59 49 10 = 766
27710 A3 61 25 6 A3 61 1A 1 20 2 = 624
27720 1C FE 69 66 69 65 6A E4 2E E9 = 1308
```

```
27730 35 2 35 6 EC 62 39 8E 6D A5 = 921
27740 EC 81 C0 20 6D 84 27 4 97 BE = 1214
27750 D7 C0 8D 7 A6 80 81 3 26 EC = 1255
27760 39 34 16 32 75 6F E4 6F 61 A7 = 1012
27770 62 E7 63 96 BE D6 C0 A7 6A 1E = 1477
27780 89 44 56 44 56 44 56 F3 55 F6 = 1173
27790 1F 1 BC 55 F8 2E 19 BC 55 F6 = 1143
27800 25 14 A6 6A 84 7 C6 80 81 0 = 923
27810 27 4 4A 54 20 F8 E7 6A EA 84 = 1184
27820 E7 84 A6 62 91 BE 25 A 90 BE = 1343
27830 A7 64 86 1 A7 66 20 A 96 BE = 1053
27840 A0 62 A7 64 86 FF A7 66 E6 63 = 1512
27850 D1 C0 25 A D0 C0 E7 65 86 20 = 1346
27860 A7 67 20 A D6 C0 E0 63 E7 65 = 1373
27870 86 E0 A7 67 A6 64 A1 65 24 12 = 1210
27880 E6 65 A7 65 E7 64 A6 66 A7 61 = 1462
27890 A6 67 A7 E4 6F 66 6F 67 A6 64 = 1357
27900 44 A7 68 86 1 A7 69 6D 66 27 = 996
27910 18 2D C 64 6A 24 12 30 1 86 = 524
27920 80 A7 6A 20 A 68 6A 24 6 30 = 743
27930 1F 86 1 A7 6A A6 E4 30 86 6C = 1123
27940 69 27 46 A6 68 AB 65 25 6 A7 = 966
27950 68 A1 64 25 24 A0 64 A7 68 6D = 1078
27960 61 27 18 2D C 64 6A 24 12 30 = 525
27970 1 86 80 A7 6A 20 A 68 6A 24 = 824
27980 6 30 1F 86 1 A7 6A A6 67 30 = 810
27990 86 BC 55 F8 22 B BC 55 F6 25 = 1256
28000 6 A6 6A AA 84 A7 84 A6 64 A1 = 1306
28010 69 24 96 32 6B 35 16 97 BE D7 = 1079
28020 C0 39 FC 55 F6 10 83 6 0 27 = 1024
28030 8 CC 6 0 8E 6D 9D 20 6 CC = 868
28040 1E 0 8E 6D A1 FD 55 F6 C3 17 = 1244
28050 FF FD 55 F8 6F 91 6F 94 16 FC = 1630
28060 5C FF CB FF CD FF CA FF CC 4A = 2000
28070 44 1 B5 44 0 B3 65 0 4C 65 = 775
```

```
10 '***** ROTBASIC *******
11 '*          Wolfgang Sohrt          *
12 '*          Borbyer Pastorenweg 3   *
13 '*          233 Eckernförde         *
20 CLEAR 200,19999:PCLEAR 8
30 GOSUB 160:REM INIT
40 CLS:PRINT"**********************
**************     3 D - R O T A
 T I O N     **********************
**************"
50 PRINT@32*4,"bitte waehlen:"
60 PRINT"(D) FIGUR DREHEN"
70 PRINT"(N) NEUE FIGUR ERSTELLE
N"
80 PRINT"(S) FIGUR AUF CASS. SPE
ICHERN"
90 PRINT"(L) FIGUR VON CASS. LAD
EN"
100 PRINT"(H) HARDCOPY-AUSDRUCK"
110 PRINT"(E) PROGRAMMENDE"
120 Q$=INKEY$:IF Q$="" THEN 120
130 Q=INSTR("DNSLHE",Q$):IF Q=0
THEN 120
140 ON Q GOSUB 1000,2000,3000,40
00,5000,6000
150 RUN
160 REM INITIALISIERUNG
170 DEF FNH(X)=INT((X+.5)/256):D
EF FNL(X)=X-FNH(X)*256+.5
180 ME=22000:DO=ME+200
190 DEF FNV(X)=X*.7-(SGN(X)=-1)*
256
200 DIM P(1):REM FUER PUT
210 RETURN
1000 PRINT:INPUT"DREHWINKEL";DW
1010 DW=DW/180*3.141592654
1020 SI=SIN(DW)*&H4000:CO=COS(DW
)*&H4000
1030 POKE ME+2,FNH(SI):POKE ME+3
,FNL(SI):POKE ME+4,FNH(CO):POKE
ME+5,FNL(CO)
1040 EXEC ME+5000
1050 RETURN
2000 I=0
2010 PMODE4,5:PCLS:PMODE4,1:PCLS
:SCREEN1,0
2020 X=0:Y=0:NR=0
2030 FLAG=0
2040 PUT(128+X,96+Y)-(128+X,96+Y
),P,NOT:PUT(128+X,96+Y)-(128+X,9
6+Y),P,NOT
2050 Q$=INKEY$:IF Q$="" THEN 204
0
2060 Q=ASC(Q$)
2070 X=X-(((Q=9)+10*(Q=93))AND X
<76)+(((Q=8)+10*(Q=21))ANDX>-76)
2080 Y=Y-(((Q=10)+10*(Q=91))ANDY
<76)+(((Q=94)+10*(Q=95))ANDY>-76
)
2090 IF Q$="L" THEN LINE(128+XA,
96+YA)-(128+X,96+Y),PSET:FLAG=0:
GOTO 2140
2100 IF Q$="S" THEN LINE(128+X,9
6+Y)-(128+X,96+Y),PSET:FLAG=1:GO
TO 2140
2110 IF Q$="E" THEN POKE DO+NR+3
,3:FOR I=0 TO 2:POKE DO+NR+I,PEE
K(DO+NR+I-4):NEXT:POKE ME,FNH(DO
+NR+4):POKE ME+1,FNL(DO+NR+4):RE
TURN
2120 IF Q$=CHR$(12) THEN IF NR>1
 THEN NR=NR-4:SOUND 200,2:SCREEN
1,1:SCREEN1,0:X=XA:Y=YA:Z=ZA
2130 GOTO 2040
2140 POKE DO+NR,FNV(X)
2150 POKE DO+NR+1,FNV(Y)
2160 PMODE 4,5:SCREEN1,0:PLAY"T2
5502CECECECEC"
```

```
2170 PUT(128+X,96+Z)-(128+X,96+Z
),P,NOT:PUT (128+X,96+Z)-(128+X,
96+Z),P,NOT
2180 Q$=INKEY$:IF Q$="" THEN 217
0
2190 Q=ASC(Q$)
2200 Z=Z-(((Q=10)+10*(Q=91))ANDZ
<76)+(((Q=94)+10*(Q=95))ANDZ>-76
)
2210 IF Q$="L" THEN LINE(128+XA,
96+ZA)-(128+X,96+Z),PSET:FLAG=0:
GOTO 2240
2220 IF Q$="S" THEN LINE(128+X,9
6+Z)-(128+X,96+Z),PSET:FLAG=1:GO
TO 2240
2230 GOTO 2170
2240 POKE DO+NR+2,FNV(Z):POKE DO
+NR+3,FLAG
2250 XA=X:YA=Y:ZA=Z
2260 NR=NR+4
2270 PMODE4,1:SCREEN1,0:PLAY"O5C
ECECECEC":GOTO 2040
3000 PRINT:INPUT"UNTER WELCHEM N
AMEN ABSPEICHERN ";N$
3010 I=DO+3
3020 IF PEEK(I)=3 THEN 3040
3030 I=I+4:GOTO 3020
3040 AUDIOON
3050 PRINT:PRINT"CASSETTE FERTIG
 MACHEN, DANN    TASTE DRUECKEN!
"
3060 IF INKEY$="" THEN 3060
3070 CSAVEM N$,ME,I,ME+5000
3080 RETURN
4000 PRINT:INPUT"WELCHES PROGRAM
M LADEN";N$
4010 AUDIOON
4020 PRINT"CASSETTE FERTIG MACHE
N, DANN    TASTE DRUECKEN!"
4030 IF INKEY$="" THEN 4030
4040 CLOADM N$
4050 RETURN
5000 PMODE4,1:SCREEN1,0
5010 Q$=INKEY$:IF Q$="" THEN 501
0
5020 IF Q$="H" THEN 5080
5030 PMODE4,5:SCREEN1,0
5040 Q$=INKEY$:IF Q$="" THEN 504
0
5050 IF Q$="H" THEN 5070
5060 GOTO 5000
5070 FOR I=1 TO 4:PCOPY I+4 TO I
:NEXT
5080 LINE(0,0)-(255,188),PSET,B
5090 LINE(0,189)-(255,191),PSET,
BF
5100 IF INKEY$="" THEN 5100
5110 POKE &H14A,2
5120 PRINT#-2,CHR$(8)
5130 EXEC 20000
5140 RETURN
6000 CLS:END
```

```
      **********************
      *   3D-ROTATION      *
      * FUER DEN DRAGON 32 *
      *WOLFGANG SOHRT, 1984*
      **********************
SPEIBG EQU   22000
       ORG   SPEIBG
       PUT   10000
DOTEND RMB   2
SIN    RMB   2
COS    RMB   2
SCRBEG RMB   2
SCREND RMB   2
MFLAG  RMB   1
F1     RMB   2
F2     RMB   2
PROD   RMB   4
DOTPTR RMB   2
XYPTR  RMB   2
XMEM   RMB   1
YMEM   RMB   1
ZMEM   RMB   2
LMNPTR FDB   NORMMT
M1LMN  FDB   $4000,0,0
       FDB   0,$4000,0
       FDB   0,0,$4000
M2LMN  FDB   $4000,0,0
       FDB   0,$4000,0
       FDB   0,0,$4000
M3LMN  FDB   $4000,0,0
       FDB   0,$4000,0
       FDB   0,0,$4000
ZWMAT  FDB   $4000,0,0
       FDB   0,$4000,0
       FDB   0,0,$4000
MABC   FDB   $4000,0,0
       FDB   0,$4000,0
       FDB   0;0,$4000
NORMMT FDB   $4000,0,0
       FDB   0,$4000,0
       FDB   0,0,$4000
NRMEND RMB   0
KEYMEM RMB   1
       ORG   SPEIBG+200
```

```
DOTBEG RMB
*** HAUPTPROGRAMM ****
       ORG   SPEIBG+5000
       PUT   11000
@START LBSR  INIT
LOOP   LBSR  TASTE
       CMPA  #3
       BNE   WEITER
       RTS
WEITER CMPA  #'R
       BEQ   @START
       LBSR  WAHL
       LBSR  DREH
       LBSR  POSI
       LBSR  DRAW
       LBSR  SWITCH
       BRA   LOOP
*xxxxxxxxxxxxxxxxxxxxx
INIT   LDX   #NORMMT
       LDY   #MABC
INITLP LDD   ,X++
       STD   ,Y++
       CMPX  #NRMEND
       BLO   INITLP
******** PMODE 4,1 ***
       LDD   #$0600
       STD   SCRBEG
       LDD   #$1DFF
       STD   SCREND
       CLRA
       STA   65472
       STA   65475
       STA   65477
       STA   65479
       LDA   #247
       STA   65314
******** PCLS ******
       LBSR  PCLS
******** DEF M.LMN **
       LDD   SIN
       STD   M1LMN+2
       STD   M2LMN+4
       STD   M3LMN+10
       COMA
       NEGB
       BCS   INIT1
       INCA
INIT1  STD   M1LMN+6
       STD   M2LMN+12
       STD   M3LMN+14
       LDD   COS
       STD   M1LMN
       STD   M2LMN
       STD   M3LMN+8
       STD   M1LMN+8
       STD   M2LMN+16
       STD   M3LMN+16
**********************
       RTS
*xxxxxxxxxxxxxxxxxxxxx
PCLS   LDX   SCRBEG
       LDD   #0
PCLSLP STD   ,X++
       CMPX  SCREND
       BLO   PCLSLP
       RTS
*xxxxxxxxxxxxxxxxxxxxx
TASTE  JSR   $8006
       BEQ   TASTE1
       STA   KEYMEM
TASTE1 RTS
*xxxxxxxxxxxxxxxxxxxxx
WAHL   LDA   KEYMEM
       LDX   #NORMMT
       CMPA  #'1
       BNE   WAHL1
       LDX   #M1LMN
WAHL1  CMPA  #'2
       BNE   WAHL2
       LDX   #M2LMN
WAHL2  CMPA  #'3
       BNE   WAHL3
       LDX   #M3LMN
WAHL3  STX   LMNPTR
       RTS
*xxxxxxxxxxxxxxxxxxxxx
DREH   LDX   #MABC
       LDY   LMNPTR
       LDU   #ZWMAT
       LBSR  MATRIX
       LDX   #ZWMAT
       LDY   #MABC
DREHLP LDD   ,X++
       STD   ,Y++
       CMPX  #MABC
       BLO   DREHLP
       LDD   #DOTBEG
       STD   DOTPTR
       LDD   #PUNBEG
       STD   XYPTR
       RTS
*xxxxxxxxxxxxxxxxxxxxx
ZCTR1  EQU   0
SCTR2  EQU   1
ELECTR EQU   2

MATRIX PSHS  D,X,Y,U
       LEAS  -3,S
       LDA   #3
       STA   ZCTR1,S
MAT1   LDA   #3
       STA   SCTR2,S
MAT2   LDA   #3
       STA   ELECTR,S
       CLR   ,U
       CLR   1,U
MAT3   LDD   ,X++
       STD   F1
       LDD   ,Y
       LEAY  6,Y
       STD   F2
       LBSR  MULS
       LDD   PROD
       ASL   PROD+2
       ROLB
       ROLA
       ASL   PROD+2
       ROLB
       ROLA
       ADDD  ,U
       STD   ,U
       DEC   ELECTR,S
       BNE   MAT3
       LEAY  -16,Y
       LEAX  -6,X
       LEAU  2,U
       DEC   SCTR2,S
       BNE   MAT2
       LEAY  -6,Y
       LEAX  6,X
       DEC   ZCTR1,S
       BNE   MAT1
       LEAS  3,S
       PULS  D,X,Y,U,
       PC

****** RTS ******
MULS   CLR   MFLAG
       TST   F1
       BGE   MULS1
       COM   MFLAG
       COM   F1
       NEG   F1+1
       BCS   MULS1
       INC   F1
MULS1  TST   F2
       BGE   MULS2
       COM   MFLAG
       COM   F2
       NEG   F2+1
       BCS   MULS2
       INC   F2
MULS2  CLR   PROD
       CLR   PROD+1
       LDA   F1+1
```

```
        LDB   F2+1
        MUL
        STD   PROD+2
        LDA   F1
        LDB   F2+1
        MUL
        ADDD  PROD+1
        STD   PROD+1
        LDA   F1+1
        LDB   F2
        MUL
        ADDD  PROD+1
        STD   PROD+1
        LDA   F1
        LDB   F2
        MUL
        ADDD  PROD
        STD   PROD
        TST   MFLAG
        BEQ   MULS3
        COM   PROD
        COM   PROD+1
        COM   PROD+2
        NEG   PROD+3
        BCS   MULS3
        LDD   PROD+1
        ADDD  #1
        STD   PROD+1
        LDA   PROD
        ADCA  #0
        STA   PROD
MULS3   RTS
*xxxxxxxxxxxxxxxxxxxx
POSI    LDX   #DOTBEG
POSILP  LDY   #MABC
        LDU   #XMEM
        LBSR  MINMAT
        LDA   ZMEM
        ADDA  #128
        PSHS  A
        LDD   XMEM
        LBSR  PERSPK
        LDY   XYPTR
        STD   ,Y++
        PULS  A
        LDA   3,X
        STA   ,Y+
        STY   XYPTR
        LEAX  4,X
        CMPX  DOTEND
        BLO   POSILP
        RTS
*xxxxxxxxxxxxxxxxxxxx
MINMAT  PSHS  D,X,Y,U
        LEAS  -3,S
MINMT1  LDA   #3
        STA   SCTR2,S
MINMT2  LDA   #3
        STA   ELECTR,S
        CLR   ,U
        CLR   1,U
MINMT3  LDA   ,X+
        STA   MF1
        LDD   ,Y
        LEAY  6,Y
        STD   MF2
        LBSR  MINMUL
        LDD   MPROD
        ASLB
        ROLA
        ASLB
        ROLA
STOP    ADDD  ,U
        STD   ,U
        DEC   ELECTR,S
        BNE   MINMT3
        LEAY  -16,Y
        LEAX  -3,X
        LEAU  1,U
        DEC   SCTR2,S
        BNE   MINMT2
        LEAS  ,S
        PULS  D,X,Y,U,PC
****** RTS ******
MINMUL  CLR   MINFLG
        TST   MF1
        BGE   MINML1
        COM   MINFLG
        NEG   MF1
MINML1  TST   MF2
        BGE   MINML2
        COM   MINFLG
        COM   MF2
        NEG   MF2+1
        BCS   MINML2
        INC   MF2
MINML2  ASL   MF2+1
        ROL   MF2
        LDA   MF2
        LDB   MF1
        MUL
        LSRA
        RORB
        STD   MPROD
        TST   MINFLG
        BEQ   MINML3
        COM   MPROD
        NEG   MPROD+1
        BCS   MINML3
        INC   MPROD
MINML3  RTS
********************
MINFLG  RMB   1
MF1     RMB   1
MF2     RMB   2
MPROD   RMB   2
*xxxxxxxxxxxxxxxxxxxx
ZORIGI  EQU   11
NEGFLG  EQU   8
YORIGI  EQU   7
XORIGI  EQU   6
YBIL    EQU   5
XBIL    EQU   4
DIVISO  EQU   2
DIVIDE  EQU   0

PERSPK  LEAS  -9,S
        STD   XORIGI,S
        LDA   #4
        LDB   ZORIGI,S
        STD   DIVISO,S
        CLR   NEGFLG,S
        LDA   XORIGI,S
        BGT   PER1
        NEGA
        COM   NEGFLG,S
PER1    CLRB
        STD   DIVIDE,S
        LDD   DIVISO,S
        ASRA
        RORB
        ASRA
        RORB
        LBSR  DIV
        TST   NEGFLG,S
        BEQ   PER2
        NEGB
PER2    ADDB  #128
        STB   XBIL,S
        CLR   NEGFLG,S
        LDA   YORIGI,S
        BGT   PER3
        NEGA
        COM   NEGFLG,S
PER3    CLRB
        STD   DIVIDE,S
        LDD   DIVISO,S
        ASRA
        RORB
        ASRA
        RORB
        LBSR  DIV
        TST   NEGFLG,S
        BEQ   PER4
        NEGB
```

```
PER4    ADDB   #128
        STB    YBIL,S
        LDD    XBIL,S
        LEAS   9,S
        RTS
*xxxxxxxxxxxxxxxxxxxxxx
DIVIDD  EQU    5
DVSOR   EQU    1
DVCTR   EQU    0


DIV     PSHS   D
        LDA    #16
        PSHS   A
        CLRA
        CLRB
        LSL    DIVIDD+1,S
        ROL    DIVIDD,S
DIVLOP  ROLB
        ROLA
        CMPD   DVSOR,S
        BLO    DIVKL
        SUBD   DVSOR,S
        ORCC   #1
        BRA    DIVWEI
DIVKL   ANDCC  #$FE
DIVWEI  ROL    DIVIDD+1,S
        ROL    DIVIDD,S
        DEC    DVCTR,S
        BGT    DIVLOP
        PULS   A
        PULS   D
        LDD    DVSOR+1,S
        RTS
*xxxxxxxxxxxxxxxxxxxxxx
DRAW    LDX    #PUNBEG
DRAWLP  LDD    ,X++
        SUBB   #32
        TST    ,X
        BEQ    DRAW1
        STA    $BE
        STB    $C0
DRAW1   BSR    LINE
        LDA    ,X+
        CMPA   #3
        BNE    DRAWLP
        RTS
*xxxxxxxxxxxxxxxxxxxxxx
XALT    EQU    10
MASKE   EQU    XALT
LINCTR  EQU    9
OFFSET  EQU    8
YINC    EQU    7
XINC    EQU    6
DY      EQU    5
DX      EQU    4
YNEU    EQU    3
XNEU    EQU    2
YALTER  EQU    1
XALTER  EQU    0

LINE    PSHS   A,B,X
        LEAS   -11,S
        CLR    XALTER,S
        CLR    YALTER,S
        STA    XNEU,S
        STB    YNEU,S
        LDA    $BE
        LDB    $C0
        STA    XALT,S
        EXG    A,B
        LSRA
        RORB
        LSRA
        RORB
        LSRA
        RORB
        ADDD   SCRBEG
        TFR    D,X
        CMPX   SCREND
        BGT    LIN0
        CMPX   SCRBEG
        BLO    LIN0
        LDA    XALT,S
        ANDA   #7
        LDB    #$80
PSEBIT  CMPA   #0
        BEQ    PSPOKE
        DECA
        LSRB
        BRA    PSEBIT
PSPOKE  STB    MASKE,S
        ORB    ,X
        STB    ,X
LIN0    LDA    XNEU,S
        CMPA   $BE
        BLO    LIN1
        SUBA   $BE
        STA    DX,S
        LDA    #1
        STA    XINC,S
        BRA    LIN2
LIN1    LDA    $BE
        SUBA   XNEU,S
        STA    DX,S
        LDA    #-1
        STA    XINC,S
LIN2    LDB    YNEU,S
        CMPB   $C0
        BLO    LIN3
        SUBB   $C0
        STB    DY,S
        LDA    #32
        STA    YINC,S
        BRA    LIN4
LIN3    LDB    $C0
        SUBB   YNEU,S
        STB    DY,S
        LDA    #-32
        STA    YINC,S
LIN4    LDA    DX,S
        CMPA   DY,S
        BHS    LIN5
        LDB    DY,S
        STA    DY,S
        STB    DX,S
        LDA    XINC,S
        STA    YALTER,S
        LDA    YINC,S
        STA    XALTER,S
        CLR    XINC,S
        CLR    YINC,S
LIN5    LDA    DX,S
        LSRA
        STA    OFFSET,S
        LDA    #1
        STA    LINCTR,S
LINLOP  TST    XINC,S
        BEQ    LIN8
        BLT    LIN9
        LSR    MASKE,S
        BCC    LIN8
        LEAX   1,X
        LDA    #128
        STA    MASKE,S
        BRA    LIN8
LIN9    LSL    MASKE,S
        BCC    LIN8
        LEAX   -1,X
        LDA    #1
        STA    MASKE,S
LIN8    LDA    XALTER,S
        LEAX   A,X
*
        INC    LINCTR,S
        BEQ    LINOUT
        LDA    OFFSET,S
        ADDA   DY,S
        BCS    LIN7
        STA    OFFSET,S
        CMPA   DX,S
        BLO    LIN6
LIN7    SUBA   DX,S
        STA    OFFSET,S
        TST    YALTER,S
```

```
        BEQ   LIN10
        BLT   LIN11
        LSR   MASKE,S
        BCC   LIN10
        LEAX  1,X
        LDA   #128
        STA   MASKE,S
        BRA   LIN10
LIN11   LSL   MASKE,S
        BCC   LIN10
        LEAX  -1,X
        LDA   #1
        STA   MASKE,S
LIN10   LDA   YINC,S
        LEAX  A,X
*
LIN6    CMPX  SCREND
        BHI   LIN12
        CMPX  SCRBEG
        BLO   LIN12
        LDA   MASKE,S
        ORA   ,X
        STA   ,X
LIN12   LDA    DX,S
        CMPA   LINCTR,S
        BHS   LINLOP
LINOUT  LEAS  11,S
        PULS  A,B,X
        STA   $BE
        STB   $C0
        RTS
*xxxxxxxxxxxxxxxxxxxxxxx
SWITCH  LDD   SCRBEG
        CMPD  #$0600
        BEQ   SWTCH1
        LDD   #$0600
        LDX   #SCR2
        BRA   SWTCH2
SWTCH1  LDD   #$1E00
        LDX   #SCR1
SWTCH2  STD   SCRBEG
        ADDD  #$17FF
        STD   SCREND
        CLR   (,X++)
        CLR   (,X)
        LBRA  PCLS
*
SCR2    FDB   $FFCB
        FDB   $FFCD
SCR1    FDB   $FFCA
        FDB   $FFCC
*xxxxxxxxxxxxxxxxxx
PUNBEG  RMB   200
        NOP
        NOP
```

Mit diesem Makro-Assembler schreiben Sie Ihre MC-Programme schnell und komfortabel

Makro-Assembler für den C-64

Vorgeschichte:

1980 hatte ich zum ersten Mal Kontakt mit Computern. Ich war fasziniert, doch erst 1984 konnte ich mir einen C-64 leisten, auf den einige Monate später eine Floppy folgte. Vom Commodore-Basic enttäuscht und alleingelassen, lernte ich Maschinensprache. Ich programmierte die Dinge, die mir fehlten, Basic-Erweiterungen, ein Musikprogramm und so weiter. Am Anfang dieses Jahres wollte ich dann ein Spiel schreiben. Ich hatte bisher mit einem Assembler gearbeitet, der einmal in einem Elektronik-Magazin abgedruckt worden war. Dieser verstand gerade einen Pseudo-Opcode, lag ausgerechnet im Bereich ab 49152 und nutzte jede Gelegenheit, um sich zu verabschieden und das Programm mit ins Grab zu nehmen, indem er den Quellcode zerstörte, ein hoffnungsloser Fall für Rettungsprogramme. Da ich selbst viel programmiere und mich daher in die Lage eines Programmierers versetzen kann, dem eines Tages eine Raubkopie seines eigenen Programmes unter der Hand angeboten wird, wollte ich mich nicht an der ewigen Raubkopiererei beteiligen und beschloß, selbst einen Assembler zu schreiben. Das klingt selbstbewußter als ich es war, nachdem ich mich mit den Schwierigkeiten dieses Unternehmens auseinandersetzte. Nach einem Monat hatte ich es dann doch geschafft, das Ergebnis folgt auf den nächsten Seiten.

Allgemeine Programm-Beschreibung

Da 6000 Datas keine Kleinigkeit sind, zuerst einige Vorzüge dieses Programms:

- **extrem hohe Arbeitsgeschwindigkeit durch eigene Auswertungsroutine: 3 kByte pro Sekunde**
- **Befehle werden tokenisiert: bis zu 30 % weniger Platzbedarf des Quelltextes gegenüber herkömmlichen Assemblern**
- **viele Programmierhilfen, z. B. Auf- und Abwärtsscrollen des Listings und Funktionstastenbelegung**
- **der Assembler liegt bis auf ca. 350 Bytes unter dem BASIC-ROM! Der Quelltext kann bis zu 38 Byte lang sein.**

Hinweise zum Abtippen

Wenn sich der BASIC-LOADER mit „assembler ok" meldet, initialisieren Sie den Assembler mit SYS49152. Es gibt nun zwei Möglichkeiten, den Assembler abzuspeichern. Wenn Sie die Funktionstasten fortwährend mit den gleichen Befehlen belegen wollen, können Sie diese Texte mitspeichern. Lesen Sie sich beim Kapitel der Tools durch, wie man Funktionstasten belegt, wählen Sie Ihre Texte und geben Sie BSAVE 43203,49520,"Name",-Gerätenummer ein. Wollen Sie die Funktionstastenbelegung nicht beibehalten, ersetzen Sie die 49520 durch 49424. Laden können sie den Assembler mit LOAD"Name",Gerätenr.,1.

Speicherbelegung

Der Assembler arbeitet im Bereich von 49203 bis 49520. Zusätzlich benutzt er den Kassettenpuffer von 828 bis 1023.

Quelltexteingabe

Nachdem Sie den Assembler gestartet haben, können Sie den Quelltext eingeben. Allerdings müssen Sie einige Besonderheiten beachten, die Sie von BASIC-Programmen nicht gewöhnt sind. Lesen Sie bitte erst die ganze Programmbeschreibung, bevor Sie ein Programm eingeben, und sehen Sie sich möglichst auch das Beispielprogramm an, selbst wenn Sie schon ein „alter Hase" sein sollten, denn das kann Ihnen viel Arbeit ersparen.

Programmbeschreibung

Am Anfang einer Quelltextzeile steht eine Zeilennummer. Dann folgt eventuell ein Label, und ab Spalte 18 steht der Befehl. Dabei gelten auch der Programmzähler und das Gleichheitszeichen als Befehle. Hinter einem Label muß ein Leerzeichen stehen. Um Speicherplatz zu sparen, eliminiert der Assembler alle Leerzeichen und tokenisiert die Befehle. Wenn Sie einen Quelltext über den Basic-Editor listen, so erscheint nur eine sinnlose Folge von Befehlen auf dem Bildschirm.

Der eigentliche Assembler

Der Assembler hat eine eigene Routine zur Auswertung von Klammerausdrücken. Der Bereich von 0 bis 65535 darf jedoch nicht überschritten werden, auch nicht bei Zwischenergebnissen. Mit den eckigen Klammern können Sie die Reihenfolge der Berechnung ändern, da der Assembler nur von links nach rechts rechnet.
Folgende Rechenarten können Sie benutzen:

Multiplikation
Addition
Subtraktion

Es gelten die normalen Rechenzeichen.
Hinzu kommen noch folgende Rechensymbole:

| OR logisches Odes
< kleiner gleich
! AND logisches Und
> größer gleich
^ EOR logisches Exklusiv-Oder

In diesen Zahlensystemen können Sie rechnen:

Binär
Kennzeichen:
% z.B. %1101
Hexadezimal
Kennzeichen:
$ z.B. $abo
Dezimal
ohne Kennzeichnung
Mit dem Zeichen ' erreichen Sie, daß die Auswertungsroutine den ASCII-Code des folgenden Buchstabens berechnet. Das Größerzeichen berechnet das Hi-Byte, das Kleinerzeichen das Lo-Byte einer Zahl. Natürlich können Sie auch Label verwenden. Der Asterisk enthält außerdem den Wert des Programmzählers. Die Befehle werden wie bei jedem anderen Assembler auch eingegeben, für die Akkumulator-Adressierung geben Sie einfach den Befehl ohne Operand ein.
Denken Sie immer daran, daß Befehle erst ab Spalte 18 stehen dürfen, da sie sonst als Label interpretiert werden! Auch ungültige Befehle verarbeitet der Assembler, als wären es Label, ohne Sie darauf aufmerksam zu machen.
Pro Zeile darf nur ein Befehl stehen, jedes Zeichen nach einem korrekt erkannten Befehl gilt als Kommentar. Kommentare können Sie einfügen, indem Sie ein Semikolon hinter den Befehl setzen, gefolgt vom kommentierenden Text.
Kommen wir nun zu einer der nützlichsten Einrichtungen eines Assemblers: den Labeln. Label sind Zeichenfolgen, denen Sie auf unterschiedliche Art Werte zuweisen können. Wie jede Assembleranweisung, so gilt auch eine Labeldefinition als Befehl und darf daher nur einmal pro Zeile erfolgen. Label sind beliebig oft redefinierbar. Stehen diese am Anfang einer Zeile, gefolgt von einem Befehl, wird ihnen der Wert des Programmzählers zugewiesen. Folgt ein Gleichheitszeichen, erhalten sie den Wert des Ausdrucks hinter dem Gleichheitszeichen. Label dürfen keine Rechensymbole enthalten, also nicht:

FORMEL+ =2

Der Assembler unterscheidet zwischen lokalen und globalen Labeln. Globale Label kann man im ganzen Quelltext verwenden. Sie werden definiert, indem man vor den Labelnamen einen Punkt setzt. Beim späteren Aufruf muß man diesen Punkt weglassen. Lokale Label definiert man ohne Zusätze. Sie gelten nur innerhalb eines bestimmten Bereiches im Quelltext. Beim Thema „Makros" wird noch näher auf diese Labelsorte eingegangen. Sie müssen sich noch merken, daß ein lokales Label nie den Namen eines globalen haben darf. Der Computer meldet sonst ein „BAD DEFINITION". Außer den Labeln kennt der Assembler noch eine andere Variable: den Programmzähler.
Der Programmzähler legt fest, wo der Assembler den Objektcode im Speicher ablegen soll. Mit BASE Ausdruck weisen Sie dem Programmzähler einen Wert zu. BASE 50000 bedeutet, daß der Objektcode ab Adresse 50000 im Speicher stehen soll. Nach der ersten Adresse kann, durch ein Komma getrennt, eine zweite Adresse folgen. Dies heißt für den Assembler, daß er einen sogenannten Relativlader erzeugen soll.
Der Assembler berechnet dann alle Adressen, Sprungbefehle usw. gemäß der ersten Adresse, legt den Objektcode jedoch beginnend bei der zweiten Adresse ab. Ein Relativlader ist z.B. nützlich, wenn Ihr Programm in einem Bereich stehen soll, der mit Quelltext belegt ist.
Nun erkläre ich Ihnen die Pseudo-Opcodes des Assemblers. Sie werden nicht assembliert, sondern weisen den Assembler an, etwas Bestimmtes zu tun. Der Befehl BASE ist ein Pseudo-Opcode, den Sie schon kennen.

Übersicht: Pseudo-Opcodes

DFB x,y,z,...
in den Objektcode werden die Bytes x,y,... eingefügt

WORD x,...
16-Bit-Werte werden im Format Lo-Hi eingefügt.

TEXT"Text"
assembliert ASCII-Codes der Zeichen, die zwischen den Anführungsstrichen stehen

GOTO x
springt zur Zeile x

END
signalisiert dem Assembler, daß der Quelltext oder eine Makro-Definition zu Ende ist.

IF Ausdruck,x
ist ein Befehl für bedingte Assemblierung. Ist der Wert des Ausdrucks gleich 0, geht der Assembliervorgang in der nächsten Zeile weiter. Ist der Wert ungleich 0, springt der Assembler zur Zeile x.

PROG"Name"
ist ein wichtiger Befehl für Floppybesitzer. Der Quellcode hinter dem PROG-Befehl wird auf Diskette assembliert.

REPEAT...UNTIL
ist eine Art umgedrehtes FOR..

NEXT
Mit diesem Befehl können Sie Assemblerschleifen bilden, allerdings keine geschachtelten. Am Beginn der Schleife steht der REPEAT-, am Ende der UNTIL-Befehl. Hinter dem UNTIL stehl ein Ausdruck. Solange er den Wert 0 hat, springt der Assembler in die Zeile hinter dem REPEAT-Befehl und assembliert weiter, bis er auf das UNTIL trifft, prüft dann den Ausdruck... usw., bis der Ausdruck ungleich 0 wird, dann assembliert er in der nächsten Zeile weiter.

Eine wichtige Funktion des Assemblers müssen Sie noch kennen: Wenn der Wert eines Ausdrucks kleiner als 256 ist, setzt der Assembler den Code für die Zero - Page - Adressierungsart ein. Manchmal muß man das vermeiden. Dann stellen Sie dem Ausdruck einfach ein Ausrufezeichen voran, z.B. LDA !100, und der Assembler setzt die absolute Adressierungsart ein.

Jetzt erkläre ich endlich die Makros. Ein Makros ist eine Befehlsfolge, die man immer wieder benötigt. Deshalb ordnet man bestimmten Programmabschnitten mit DMACRO einen Namen zu. Die Definition beenden Sie mit dem END-Be-

fehl. Vor dem Ende des Programms müssen alle Makrodefinitionen abgeschlossen sein. Innerhalb von Makros gelten die gleichen Regeln wie für den „normalen" Quelltext. Die Makrodefinition selbst wird nicht assembliert, dazu müssen Sie das Makro mit GMACRO Name aufrufen. Makros dürfen bis zu 25-mal geschachtelt werden. Insgesamt dürfen Sie 254 Makros aufrufen, eine sehr theoretische Begrenzung, denn selbst beim 35kByte langen Quelltext dieses Assemblers werden Makros höchstens 50-mal aufgerufen.
Makros funktionieren ähnlich wie der Befehl GOSUB in BASIC. Der Assembler merkt sich die Zeile, in der das Makro aufgerufen wurde, geht zu der Zeile der Makro-Definition, assembliert den dort stehenden Quelltext, bis er auf die END-Anweisung trifft und macht dann in der ersten Zeile nach dem Makro-Aufruf weiter.
Es besteht die Möglichkeit, Werte an das Makro zu übergegen, ohne dafür globale Label zu verwenden. Dazu setzt man bei der Makro-Definition hinter den Makronamen Label, die bei jedem Aufruf neu definiert werden, und zwar mit den Parametern hinter dem Makroaufruf. Das sieht so aus:

```
DMAC  SETVEC,ADRESSE,
      VECTOR
LDA   ADRESSE
LDX   ADRESSE
STA   VECTOR
STX   VECTOR+1
END

.(anderer Quelltext)

.
GMAC SETVEC,49152,768
```

gibt den Objektcode:

```
LDA 0      STA 768
LDX 192    STX 769
```

Die Anzahl der Parameter muß mit der Anzahl der Label in der Makro-Definition übereinstimmen. Bei Fehlern, die in der DMAC-Zeile auftreten, wird als Fehlerzeile die Nummer der GMAC-Zeile ausgegeben.
Makro-Parameter werden zu lokalen Labeln und können nur innerhalb des Makros verwendet werden. Der Assembler weist ihnen bei jedem Aufruf eine Nummer zu, die sie von anderen Labeln unterscheidet. Auch Label aus verschiedenen Makros, die einen gleichen Namen tragen, kann der Assembler auseinanderhalten.

Tools

Über den Assembler wissen Sie jetzt alles. Damit Ihnen das Programmieren etwas leichter fällt, habe ich noch einige Befehle eingebaut, die nur im Direktmodus eingesetzt werden können. Diese müssen nicht formatiert eingegeben werden. Der Assembler führt nur den ersten Befehl einer Zeile aus. Wie in BASIC können Sie die Befehle abkürzen. Außer diesen Befehlen helfen noch einige Tastenfunktionen. Nach SHIFT+CONTROL tabuliert der Cursor zur Spalte 18, wenn Sie nochmals drücken, wieder zurück zur Spalte 6. Die Funktionstasten 1-6 können Sie belegen, 7 und 8 scrollen das Listing auf- und abwärts. Doch nun zu den Befehlen:

KEY n,"Text"
belegt Funktionstaste von 1-6 mit einem Text, der höchstens 13 Zeichen lang sein darf. Der Pfeil nach rechts steht für ein Carriage Return. Sie können mit den Funktionstasten z. B. das Listing in Teile gliedern, die Sie immer wieder brauchen, indem Sie den LIST-Befehl mit verschiedenen Zeilennummern auf den Funktionstasten ablegen.

ASM x
assembliert Quelltext ab Zeile x

DISPLAY
zeigt Funktionstastenbelegung

AUTO x,y
automatische Zeilennummerausgabe, beginnt bei x, Schrittweite y. Ein Befehl im Direktmodus stoppt die Nummerierung.

RENUM x,y,z
nummeriert ab jetziger Zeile x, beginnt mit Zeilennummer y, Schrittweite z

POKE x,y
bei Werten kleiner 256 wie in Basic, bei größeren Werten wird zusätzlich das Hi-Byte abgelegt.

MEM
zeigt Speicherbelegung

PEEK/DEEK x
zeigt ein- bzw. zwei-Byte-Adressen

? Ausdruck
wertet den Ausdruck hinter dem Fragezeichen aus und gibt das Ergebnis aus

CLR
löscht Variablen

LIST
formatiertes Listen

NEW
löscht Programm

DEL
löscht Programmbereiche.
Format: DEL x-y , DEL -x,DEL x-
x und y sind Zeilennummern. Gelöscht wird einschließlich der angegebenen Zeilen.

DUMP
gibt die verwendeten Label aus. Makronamen werden ebenfalls ausgegeben, sie enthalten die Adresse, ab der das Makro als Quellcode im Speicher steht.

OLD
rettet gelöschtes Programm. Es kann vorkommen, daß in der ersten Zeile ein Byte geändert wird. Da dies aber in der Regel der Programmzähler ist, läßt sich der Fehler problemlos überschreiben, und das Programm ist danach wieder voll lauffähig.

HUNT
Nachdem Sie diesen Befehl eingegeben haben, erscheint eine Zeile darunter der blinkende Cursor. Jetzt können Sie einen Text eingeben, nach dem Sie suchen wollen. Wenn Sie nach einem Label suchen, geben Sie Anführungsstriche und danach den Labelnamen ein. Einen Befehl findet die Routine nur, wenn Sie ihn ab Spalte 18 ohne Anführungsstriche eingeben. Sie können bei der Labelsuche „Joker" benutzen. Ein Fragezeichen steht für einen unbekannten Buchstaben, ein Asterisk (*) für den unbekannten Rest eines Labels. Alle Zeilen, in denen das Gesuchte steht, werden ausgegeben.

QUIT
Der Assembler wird verlassen.

BSAVE x,y,
„Name", Gerätenummer
speichert den Bereich zwischen x und y.
Der untere ROM-Bereich wird vorher abgeschaltet.

BLOAD x,
„Name", Gerätenummer
lädt ein Programm an die Adresse x.

MERGE „Name",
Gerätenummer
lädt Quelltext nach und fügt ihn an den bestehenden Quelltext an.

Diskettenbefehle

Klammeraffe
zeigt den Fehlerkanal.

SEND"Text"
sendet Text zur Floppy.

CAT
liest Directory ohne Programmverlust ein.

FAST
vermindert die Zugriffszeit des Lesekopfes. Das angsteinflößende Knattern beim VALIDATE-Befehl wird auf ein Minimum herabgesetzt.

DEV n
legt die Gerätenummer der Floppy fest.

Der Assembler holt sich beim ersten Start die Gerätenummer des angeschlossenen Gerätes, von dem er geladen wurde. Stellt er die Gerätenummer der Datasette fest, antwortet er auf die Disk-Befehle nur mit einem READY. Nach einem Kaltstart kann dies passieren, obwohl eine Diskettenstation angeschlossen ist. Nach einem DEV

n ist dieser Fehler behoben.
Für die Druckausgabe steht der Befehl OUT n, Zeilenbereich zur Verfügung. Ein Listing wird an das Gerät mit der Gerätenummer x gesandt. Der Zeilenbereich wird wie beim LIST-Befehl angegeben.
Vielleicht wundern Sie sich, daß die Befehlserweiterung keine Standardbefehle wie LOAD oder SAVE enthält.
Da der Assembler unter dem ROM liegt, treten einige Schwierigkeiten auf, wenn man auf die ROM-Routinen zugreifen will. Weil man Befehle wie SAVE relativ selten gebraucht (wenn Sie das Programm nach längerer Arbeit abspeichern), und es noch einige DATAS mehr erfordert hätte, diese Befehle einzubauen, habe ich darauf verzichtet. Sie können die Befehle natürlich benutzen, wenn Sie den Assembler mit QUIT verlassen.

Sie wissen jetzt, wie der Assembler funktioniert. Es kann natürlich vorkommen, daß etwas mit Ihrem Programm nicht stimmt. Um Ihnen die Fehlersuche zu erleichtern, hat der Assembler eine Reihe eingebauter Fehlermeldungen.
Neben den BASIC-Meldungen können folgende Fehler auftreten:

UNDEFD LABEL
ein Label wurde noch nicht definiert

UNKNOWN CODE
ein Befehl konnte nicht erkannt werden, oder die Adressierungsart ist unzulässig.

BAD BRANCH
die Grenzen eines relativen Sprungbefehls wurden überschritten.

STACK FULL
es wurden zu viele Makros benutzt (mehr als 254) oder sie wurden zu tief geschachtelt

BAD MAECRO NAME
ein Makro ist nicht definiert

BAD LOOP
ein REPEAT oder UNTIL zu viel oder zu wenig ILLEGAL PARAMETER verschiedene Parameteranzahl zwischen Definition und Aufruf

END OF LINE EXPECTED
statt eines Zeilenendes fand der Assembler ein anderes Zeichen

DEFINITION NOT FINISHED
der Quelltext ist zu Ende, bevor alle Makro-Definitionen geschlossen wurden.

OPEN MACROS
innerhalb eines Makros wurde das Quellentextende gefunden.

BAD DEFINITION
ein lokales Label sollte definiert werden, es war aber schon ein globales Label mit gleichem Namen vorhanden oder umgekehrt.

Ende der Programmbeschreibung

Thorsten Engel

Wir suchen Programmautoren für folgende Computersysteme: Commodore, MSX, Atari, Schneider, und andere. – Wir zahlen 120,– DM pro veröffentlichte Seite!

C64 LISTING MIT CHECKSUMMEN (C V1.0)

```
0 A=43203:EP=49427                                                    <26>
1 P=0:READA$:IFLEN(A$)<>26THEN5                                       <173>
2 ON-(VAL(A$)<0)GOTO6:FORG=1TO26STEP3:W=
(ASC(MID$(A$,G))-65)*16                                               <76>
3 W=W+ASC(MID$(A$,G+1))-65:P=P+W:POKEA,W
:A=A+1:H=H+W:NEXT:H=H-W                                               <78>
4 H=H+65535*(H>65535):P=P-W:A=A-1:IF (PA
ND255)=W THEN 1                                                       <59>
5 PRINT"PRUEFSUMMENFEHLER{SPACE}IN"PEEK(
63)+PEEK(64)*256:END                                                  <57>
6 IF H<>18564THEN PRINT"GESAMTSUMME{SPAC
E}FALSCH":END                                                         <154>
7 IF A<EPTHEN PRINT"ZU{SPACE}WENIGE{SPAC
E}ZEILEN":ENDEADY.                                                    <115>
8 IF A>EPTHEN PRINT"ZU{SPACE}VIELE{SPACE
}ZEILEN":END                                                          <101>
9 IF A=EPTHENPRINT"ASSEMBLER{SPACE}OK"                                <203>
100 DATA AA AA AA PP AB AA AA AA AA                                   <238>
101 DATA AB AI JJ BB AB BK AA AB MP                                   <53>
102 DATA JI BI GF HK IF HK JA AC CA                                   <54>
103 DATA OG HL GA EM CP MA KA AA JM                                   <115>
104 DATA IE FA IE FB LB HK DI OJ PF                                   <108>
105 DATA DA JA OF MJ AK LA OB IF IO                                   <111>
106 DATA DM KF FB IF DO KF FA IF GP                                   <252>
107 DATA DN AK CG FB AK CG FB GF KE                                   <143>
108 DATA DN IF FA KF FB GF DO IF DA                                   <186>
109 DATA FB AG FA CG FB LA MM KF DP                                   <225>
110 DATA FA GF DM IF FA JA AE OG EA                                   <100>
111 DATA FB PA MA IF FA MI NA ME DC                                   <45>
112 DATA GA AA AB AC AD AE AF AG HF                                   <248>
113 DATA AH AI AJ PP PP PP PP PP BD                                   <11>
114 DATA PP PP AK AL AM AN AO AP EJ                                   <94>
115 DATA KC AA IG FA IG FB MJ DA EI                                   <169>
116 DATA JA CN OJ DA MJ BH LA CH IN                                   <244>
117 DATA KK LN CE KJ DA CB IF DM EG                                   <17>
118 DATA KF FB BA AD EM CP MA KF OJ                                   <30>
119 DATA FA AK CG FB AK CG FB AK FM                                   <171>
120 DATA CG FB AK CG FB AF DM IF LO                                   <180>
121 DATA FA CA HD AA EM EB KJ GA HJ                                   <207>
122 DATA KC AA IG FA IG FB LA KI KH                                   <108>
123 DATA MJ DC LA CF KE FB BA AD NI                                   <95>
124 DATA EM CP MA EK CG FA CG FB HC                                   <74>
125 DATA CA HD AA EM HJ KJ BI KF LO                                   <153>
126 DATA EO GF FA IF EO KF EP GF CP                                   <118>
127 DATA FB IF EP JA AD EM CP MA PD                                   <255>
128 DATA GA EM MO MA DI KF EO OF EK                                   <214>
129 DATA FA IF EO KF EP OF FB IF NC                                   <43>
130 DATA EP JA AB GA KF DL PA PL AL                                   <46>
131 DATA EM CP MA EM MO MA CA BF EK                                   <127>
132 DATA LB PA PI CA HJ AA LA AD OF                                   <204>
133 DATA EM OB KI MJ ML NA AP CA GI                                   <37>
134 DATA HD AA CA PN LC KJ AA IF HA                                   <218>
135 DATA FB KF EO IF FA GA MJ MH AJ                                   <199>
136 DATA PA GE MJ MM NA AP CA HD FL                                   <212>
137 DATA AA CA PN LC KF EP IF FA JI                                   <3>
138 DATA KJ AA IF FB GA MJ CF PA LN                                   <18>
139 DATA AK MJ CE NA AM CA HD AA GG                                   <31>
140 DATA EM DL KJ CA HD AA EM HD IC                                   <178>
141 DATA KJ MJ CH NA AN KA AA IE JK                                   <105>
142 DATA FB MI LB HK IF FA MI EM CN                                   <88>
143 DATA ND KI CA HI KL CA DK KL MD                                   <47>
144 DATA JA AM KA AC LB DP IF FA AD                                   <80>
145 DATA MI LB DP IF FB GA KN MG GB                                   <211>
146 DATA KI MJ AC JA AD EM AC KN AB                                   <140>
147 DATA KA AA IE FA IE EO IE EP BJ                                   <165>
148 DATA IE FB MI IE FG GA KF PL HH                                   <110>
149 DATA IF FA KF PM IF FB EM HD AL                                   <187>
150 DATA AA KF EO IF FA KF EP IF EB                                   <114>
151 DATA FB GA CA AN LN KF EP KG DF                                   <39>
152 DATA EO CA EH MA CA AN LN CA HP                                   <208>
153 DATA EM LD CA AN LN CA GA LH CA                                   <181>
154 DATA EM HB MA KF FA IF EO KF OK                                   <184>
155 DATA FB IF EP GA KF EO MF FA IN                                   <93>
156 DATA KF EP OF FB KC AA KA AB GN                                   <188>
157 DATA GA CA HP KK JA AB II IG EI                                   <131>
158 DATA EP IE EO GA CA HP KK LA HK                                   <106>
159 DATA PG JA PD CA IH KK KF EO LN                                   <75>
160 DATA MF FA NA OK KF FB MF EP NJ                                   <30>
161 DATA NA OE PA OD KJ BA IF DM AB                                   <9>
162 DATA KC AA KA AA IK AK KK JI BI                                   <156>
```

```
163 DATA CK KI JA AD EM CP MA AG KG      <161>
164 DATA FA CG FB JA AL BI IK GF GJ      <128>
165 DATA EO KK JI GF EP KI LA OM II      <151>
166 DATA MG DM NA OA IG EO IE EP FJ      <190>
167 DATA GA EM MO MA DI OJ MF KK MK      <239>
168 DATA LN PJ KK IF EH LN PA KK ID      <68>
169 DATA IF EI GM EH AA KJ KJ KK HM      <3>
170 DATA LD LD LD KK KK KK JB KH EP      <90>
171 DATA KP GL HI IF IM JH JO KF HN      <223>
172 DATA CI KK KA AA JB AC KF AC KM      <238>
173 DATA DI OJ AF LA AD MG AD DI NK      <183>
174 DATA OF CI IF DP JF AC KF AD AA      <38>
175 DATA OJ AA IF AD IF EA KJ PP NO      <129>
176 DATA JB AC MI JB AC MI KJ AA FP      <184>
177 DATA JB AC MI JB AC MI LJ DI KH      <65>
178 DATA AD JB AC MK BA PH GA KF GM      <222>
179 DATA PN IF DP KF PO IF EA KA MJ      <213>
180 DATA AA LB DP DA CO KK BI GJ HJ      <72>
181 DATA AF IF DM DI KF DP OF DM AD      <9>
182 DATA IF DP LA AC MG EA OE CI II      <186>
183 DATA NA OH MI LB DP PA AF MN DB      <53>
184 DATA NC KI NA NL KA AD MI LJ EJ      <168>
185 DATA DI AD NB DP NA NB MK BA MG      <29>
186 DATA PF DI GA BI GA KA PP MI GM      <14>
187 DATA LB HK JJ DM AD PA AC BA AF      <253>
188 DATA PG JI BI GF HK IF HK JA BE      <142>
189 DATA AC OG HL II IE CI DA AB MI      <63>
190 DATA GA EM MO MA KF PN IF DP KA      <54>
191 DATA KF PO IF EA KA AA LB DP PI      <57>
192 DATA DA EL KK BI GJ AF IF DM GM      <70>
193 DATA KF DP DI OF DM IF DP LA LB      <121>
194 DATA AC MG EA KA AB LB DP MJ GC      <144>
195 DATA AC MI LA OA LB DP IF EO BN      <165>
196 DATA MI LB DP IF EP MI LB DP EE      <190>
197 DATA CA NC PP MI MK BA PH CA KK      <177>
198 DATA AN LN MI MA BC JA PI KF JB      <58>
199 DATA EP KG EO CA EH MA CA AN JH      <113>
200 DATA LN CA HB MA CA FG MA CA GE      <60>
201 DATA BC LN EM JP KL GA EM MO NP      <191>
202 DATA MA KA AA LB HK PA DA MJ HE      <160>
203 DATA ME PA CM MJ CI PA CL MJ LF      <27>
204 DATA CB PA FO MJ CD PA EF CA LA      <18>
205 DATA LO LC CA HJ AA MJ NB PA JD      <237>
206 DATA AI MJ NA PA AL KC AC NA BA      <148>
207 DATA EA CA HG LC KC AE NA DJ DH      <5>
208 DATA CA HG LC KC AG NA DC KC JE      <22>
209 DATA AA GA CA HG LC CA LO LC DI      <139>
210 DATA CA HJ AA MJ MP PA AO MJ PI      <242>
211 DATA MO NA LD CA HG LC KC AJ EE      <45>
212 DATA NA BC EM CP MA CA HD AA LA      <100>
213 DATA KC AI NA AI CA HD AA CA DF      <113>
214 DATA LO LC KC AB KF EP NA OK MB      <216>
215 DATA GA KF EP AF FG PA AB OI II      <25>
216 DATA GA CA HD AA CA AK KM KF GO      <236>
217 DATA EP AF FG NA PD OI GA EM AB      <145>
218 DATA KI MA KA AA LB HK MJ EB DN      <238>
219 DATA JA BB MJ FL LA AN CA HI BK      <131>
220 DATA KL CA DK KL LA AE CA AC IG      <134>
221 DATA KL BI GA EM MO MA KG AD KG      <79>
222 DATA MK OE CO JA NK KA AA LB JH      <230>
223 DATA HK MJ CO PA EK CA KP KM CG      <235>
224 DATA KA AB LB DP PA FB KN NC FB      <238>
225 DATA KI JB DP GA CA HF KM KA LJ      <91>
226 DATA AA LB HK MJ MN NA CE KF FK      <188>
227 DATA DP EI KF EA EI CA HD AA EH      <65>
228 DATA CA LO LC GI IF EA GI IF KK      <108>
229 DATA DP CA NC KM EM JI LP KA CA      <119>
230 DATA AD KF EP JB DP II KF EO EC      <128>
231 DATA JB DP GA KA AD KF PM JB AF      <167>
232 DATA DP II KF PL JB DP GA CA LH      <40>
233 DATA HD AA CA KP KM KA AB LB EA      <219>
234 DATA DP PA AH MJ PP PA AD EM DN      <16>
235 DATA AL KN KJ AA JB DP GA KC DD      <65>
236 DATA AA CM KC AN CM KC BJ CM OO      <32>
237 DATA KC CD CM KC DL CM KC DB MN      <101>
238 DATA CM KC EJ CM KC GF CM KC BI      <218>
239 DATA FE CM KC HJ IG DN CA MM EK      <169>
240 DATA PP KJ AA IF BD CA HB MA JB      <166>
241 DATA KJ DP CA AM OB KG DN LN JF      <71>
242 DATA EF KN EI CJ HP CA AM OB OP      <100>
243 DATA OI GI BA PD CA GP LF EM OD      <197>
244 DATA NE MA FF EO EE EF EG CH CN      <178>
245 DATA EE CA EM EB EC EF MM FF JJ      <35>
246 DATA EO EL EO EP FH EO CA ED DO      <68>
247 DATA EP EE MF EC EB EE CA EC IB      <37>
248 DATA FC EB EO ED MI EC EB EE LD      <54>
249 DATA CA EE EF EG EJ EO EJ FE CD      <83>
250 DATA EJ EP MO FD FE EB ED EL NM      <174>
251 DATA CA EG FF EM MM EC EB EE JK      <235>
252 DATA CA EN EB ED FC EP CA EO AA      <176>
253 DATA DA EB EN MF EP FA EF EO CA KF      <203>
254 DATA EN EB ED FC EP ND EJ EM NK      <44>
255 DATA EM EF EH EB EM CA FA EB BG      <37>
256 DATA FC EB EN EF FE EF NC EF NF      <231>
257 DATA EO EE CA EP EG CA EM EJ PM      <196>
258 DATA EO EF CA EF FI FA EF ED CI      <175>
259 DATA FE EF ME EC EB EE CA EM JA      <186>
260 DATA EP EP NA EB EE MD EB EO EF      <179>
261 DATA ME EB FD MM EC EJ NE ED MG      <80>
262 DATA EN NA ED FA NI ED FA NJ PE      <121>
263 DATA EE EF MD EF EP NC EJ EO EJ      <210>
264 DATA MD EM EE MB EM EE NI EM MI      <77>
265 DATA EE NJ EM FD NC EP FC MB PA      <124>
266 DATA FC EP MM FC EP NC FD EC HF      <1>
267 DATA MD FD FE MB FD FE NI FD PN      <236>
268 DATA FE NJ EC FC ML ED EM MD NO      <127>
269 DATA ED EM ME ED EM MJ ED EM DK      <160>
270 DATA NG EE EF NI EE EF NJ EJ OC      <199>
271 DATA EO NI EJ EO NJ FA EI MB OP      <206>
272 DATA FA EI NA FA EM MB FA EM GB      <99>
273 DATA NA FC FE MJ FC FE ND FD AL      <164>
274 DATA EF MD FD EF ME FD EF MJ MF      <131>
275 DATA FE EB NI FE EB NJ FE FD IC      <144>
276 DATA NI FE FI MB FE FI ND FE BI      <209>
277 DATA FJ MB EO EP NA EC ED MD MP      <42>
278 DATA EC ED ND EC EF NB EC EN DP      <39>
279 DATA MJ EC EO MF EC FA MM EC LO      <244>
280 DATA FG MD EC FG ND EK EN NA OL      <101>
281 DATA EK FD NC EF EO ME EE EN FH      <130>
282 DATA EB MD EH EN EB MD EE EG CG      <93>
283 DATA MC FE EF FI NE FH EP FC HP      <248>
284 DATA ME EH EP FE MP EJ MG EC MO      <201>
285 DATA EB FD MF FC EF FA EF EB MG      <42>
286 DATA NE FF EO FE EJ MM FA FC IC      <221>
287 DATA EP MH LL KL KN KK CB FI EM      <134>
288 DATA EP NC CB EB EO ME CB EP AF      <159>
289 DATA NC LM LO LN CJ CM NJ CM GD      <148>
290 DATA FI KJ CM NJ CM NI KJ KM FP      <197>
291 DATA NL NN AA FB FF EJ NE EM MH      <238>
292 DATA EJ FD NE EB FD MN ED EM GA      <231>
293 DATA NC EE FF EN NA EF EO ME NP      <200>
294 DATA FA EP EL MF LP EN EF MN MN      <217>
295 DATA EB FF FE MP EL EF NJ FC HE      <140>
296 DATA EF EO FF MN EP EM ME EO GC      <105>
297 DATA EF NH EI FF EO NE ED EB FP      <124>
298 DATA NE EE EJ FD FA EM EB NJ GK      <103>
299 DATA EC FD EB FG MF EC EM EP MO      <96>
300 DATA EB ME MA EN EF FC EH MF LF      <105>
301 DATA FD EF EO ME EE EF MM FA EP      <190>
302 DATA EF EF ML EE EF EF ML EG DE      <13>
303 DATA EB FD NE EE EF NG EP FF GL      <208>
304 DATA NE AA AB AC AC AD AC AD OB      <141>
305 DATA AC AD AC AC AA GJ GF GN EE      <20>
306 DATA HF HN AA HJ GB HB AA CJ GG      <245>
307 DATA CF CN DF DN AA DJ CB DB EP      <60>
308 DATA AK AA AG AO BG BO AA AA FC      <9>
309 DATA AA AA AA AA CE CM AA AA FA      <4>
310 DATA AA AA AA AA AA MJ MF MN FL      <119>
311 DATA NF NN AA NJ MB NB AA OA PN      <32>
312 DATA OE OM AA AA AA AA AA AA NA      <63>
313 DATA AA MA ME MM AA AA AA AA FA      <116>
314 DATA AA AA AA AA MG MO NG NO EI      <105>
315 DATA AA AA AA AA AA EJ EF EN NL      <188>
316 DATA FF FN AA FJ EB FB AA AA JN      <185>
317 DATA OG OO PG PO AA AA AA AA MI      <84>
318 DATA AA KJ KF KN LF LN AA LJ CG      <233>
```

```
319 DATA KB LB AA KC KG KO AA AA EI    <30>
320 DATA LG LO AA AA AA KA KE KM GE    <191>
321 DATA LE LM AA AA AA AA EK AA LK    <158>
322 DATA EG EO FG FO AA AA AA AA EI    <1>
323 DATA AA AJ AF AN BF BN AA BJ GG    <6>
324 DATA AB BB CK AA CG CO DG DO AE    <53>
325 DATA AA AA AA AA GK AA GG GO DO    <172>
326 DATA BC HO AA AA AA AA AA OJ HJ    <211>
327 DATA OF ON PF PN AA PJ OB PB IP    <128>
328 DATA AA AA IF IN JF JN AA JJ NN    <129>
329 DATA IB JB AA AA IG IO AA AA CG    <68>
330 DATA JG AA AA AA AA AA IE IM KG    <13>
331 DATA JE AA AA AA AA AA IG LG NA    <180>
332 DATA IG LG IG LG IG LG IG LG PA    <115>
333 DATA IG LG IG LG IG LG FD LK MB    <62>
334 DATA EE LK FC LG HL LK NP LK NE    <231>
335 DATA BJ LN EK LN MG LO PL LO BK    <94>
336 DATA MO MA PG LB FI LP GM LP HH    <179>
337 DATA KA LP KD KJ PC LA JM LH KA    <140>
338 DATA BC LF CD LD JH KL FC LG OH    <93>
339 DATA FK LG IA LG BP LH HG LH EJ    <104>
340 DATA DJ LJ HI LL MF LL JD MA PI    <63>
341 DATA NO LL GD LM NF LM HH LN HN    <32>
342 DATA FG LN NG LN KK LN PP LN MJ    <103>
343 DATA GL LO NL LO OJ LO DI LP GA    <206>
344 DATA MM LM BJ LP EM AF KN MJ CH    <223>
345 DATA BM LA PJ AK KK LN CP LA BF    <60>
346 DATA IF EH LN DA LA IF EI CA FG    <141>
347 DATA HD AA GM EH AA EM KP LC ND    <248>
348 DATA JI EM FH LC CJ HP KI MJ AG    <105>
349 DATA EF LA NJ DI OJ BF JA PA IE    <148>
350 DATA KK OJ BJ JA OI KI AK KK IA    <55>
351 DATA LN AB LA IF EH LN AC LA KJ    <160>
352 DATA IF EI JI GM EH AA KA AA LI    <161>
353 DATA IM HE LH LB HK BA LF CJ NA    <98>
354 DATA HP BA LE KJ MA KC KO IN IJ    <33>
355 DATA AD AD IO AC AD CA NI LA EB    <18>
356 DATA KJ AO IN IJ AC CA OF LA IE    <63>
357 DATA KN MM KI NA AF KF LK IN OC    <8>
358 DATA MM KI EM IN MA HI KJ MA OO    <161>
359 DATA KC GC IN JA AC IO IP AC EC    <4>
360 DATA FI GA KJ MB KC KI IF PN OO    <239>
361 DATA IG PO IF AC IG AD GA CA BE    <126>
362 DATA FD OE CA BF PN GI GI KJ OC    <135>
363 DATA KA KC AA IF DI IG DH CA NM    <68>
364 DATA AI LB EM IN MA KJ OL KC II    <15>
365 DATA EI HI IN JA AC IO IP AC PO    <220>
366 DATA FI GA KA AA LB HK PA AE HH    <205>
367 DATA MJ ME PA AA GA DI OJ HP HN    <184>
368 DATA EI KK IE EJ KM NA KI MA KD    <93>
369 DATA BB LA AN KJ CA CA NC PP II    <54>
370 DATA MI MA BB JA PI IM NA KI CF    <243>
371 DATA KA PP MK PA AI MI LJ MG KI    <120>
372 DATA KN BA PK DA PF MI LJ MG CD    <85>
373 DATA KN DA AJ OO NA KI CA NC DO    <26>
374 DATA PP EM EI LB CJ HP OO NA KK    <165>
375 DATA KI CA NC PP GI MJ EF LA LP    <54>
376 DATA AF KJ CA CA NC PP KE EJ KM    <193>
377 DATA OO NA KI GA EM MO MA CA MA    <170>
378 DATA HJ AA JA AG PA AE MJ MG JC    <119>
379 DATA NA PC CA ON MA CA HK MA OJ    <60>
380 DATA CA HJ AA PA AM MJ MG NA PE    <119>
381 DATA OD CA HD AA CA ON MA NA BD    <146>
382 DATA NL KF BE AF BF NA AG KJ CN    <85>
383 DATA PP IF BE IF BF GA KA AA DC    <132>
384 DATA KN MG KI PA AP LB HK KK OP    <7>
385 DATA DA BO CA JB KM KA AA LB PM    <194>
386 DATA HK DA BF GA LB HK DA AD HN    <181>
387 DATA CA HI KL KA AA LB HK MJ NH    <118>
388 DATA LJ PA AL MJ LI PA AH GA IM    <39>
389 DATA KO MG KI MK PA AD EM IH KM    <248>
390 DATA LA MJ ME PA BN MJ JF JA DI    <27>
391 DATA PF MJ KO JA AL MJ LG JA BG    <254>
392 DATA AK MJ LI JA AJ EM IH LA KH    <55>
393 DATA KJ AB CM KJ AC CM KJ AD FJ    <84>
394 DATA IF CJ GA CA HG LC CA LO DE    <245>
395 DATA LC KG EO IG PL KF EP IF KA    <200>
396 DATA PM KA AA LB HK MJ ND NA DD    <179>
397 DATA BL CA HD AA KF DL EI OG LM    <102>
398 DATA DL CA LO LC GI IF DL KF JI    <127>
399 DATA PL DI OF EO IF IL KF PM BH    <56>
400 DATA OF EP IF IM KM MP KI NA DI    <115>
401 DATA BA CA FE LO KF AE KG AF JG    <74>
402 DATA IN EM LG IO EN LG OO MP NN    <127>
403 DATA KI GA EM MO MA AA AK BE AA    <4>
404 DATA BO CI DC DM EG FA FK GE AI    <211>
405 DATA GO HI IC IM JG KA KK LE II    <204>
406 DATA LO MI NC NM KK LN EA LC IN    <111>
407 DATA EI CA HG LC CA PE KL IG NF    <90>
408 DATA DM GI BI GF DM KI LJ CP ON    <211>
409 DATA KP PA AP IF EN LN CF KP BB    <124>
410 DATA IF CJ GA OG HK NA AC OG CG    <131>
411 DATA HL GA OA AC PA AL OA AE JM    <124>
412 DATA PA AH OA AG PA AD EM AF CB    <29>
413 DATA KN OI MI NA NJ CA LO LC JG    <240>
414 DATA GM EO AA AA BI NI FI LI LK    <133>
415 DATA MK II OI MI EI AI GI CI OC    <80>
416 DATA EA GA DI PI HI KK KI LK FE    <231>
417 DATA IK JK JI OK LN JG LC IF DA    <88>
418 DATA EN KJ AB IF CJ EM HG LC BJ    <169>
419 DATA EM MO MA KJ AA IF EB IF MO    <78>
420 DATA FG CA PN LC CA BF LB PA PL    <65>
421 DATA CJ MJ NF PA EL MJ MF JA CA    <216>
422 DATA OH MJ MO LA BN EI CA HG CJ    <169>
423 DATA LC KF EO EI KF EP EI CA EJ    <184>
424 DATA PN LC CA FE KK GI IF EP AJ    <35>
425 DATA GI IF EO GI CA NP KK EM JI    <84>
426 DATA MH LC KF EB NA AB GA EM NM    <211>
427 DATA MO MA KA AA LB HK MJ NE PG    <236>
428 DATA NA AI CA HD AA OG EB EM NO    <11>
429 DATA ME LC CA MB KJ CA HG KK EA    <42>
430 DATA KA AA LB HK MJ NF NA AH EA    <75>
431 DATA CA HD AA MG EB DA NI GA AC    <0>
432 DATA KJ KI KC MB IG PN IG AC LP    <33>
433 DATA IF PO IF AD KA AA KJ IB NF    <146>
434 DATA JB AC GA EI EK EK EK EK GD    <131>
435 DATA KK LN FL LD CA AM OB GI OK    <214>
436 DATA CJ AP KK LN FL LD EM AM AF    <91>
437 DATA OB KJ CE CA AM OB KF EP KP    <2>
438 DATA CA DG LD KF EO EM DG LD DB    <119>
439 DATA DA DB DC DD DE DF DG DH JM    <2>
440 DATA DI DJ EB EC ED EE EF EG AG    <41>
441 DATA KF EO EF FA IF EO KF EP EP    <254>
442 DATA EF FB IF EP GA KF EO CF OC    <251>
443 DATA FA IF EO KF EP CF FB IF BC    <254>
444 DATA EP GA KF EO AF FA IF EO MK    <121>
445 DATA KF EP AF FB IF EP GA IG AE    <16>
446 DATA HK IE HL KA PP MK MI OI JC    <3>
447 DATA LN AA AC MJ CA PA PI DI MI    <118>
448 DATA PB DN PA PC MJ IA NA AH DA    <49>
449 DATA AN NB KI KE HL DI GA LB OO    <240>
450 DATA DN PA AM DA AD MI NA PH PL    <45>
451 DATA KG HK OO NB KI NA NG KG ND    <112>
452 DATA HK KE HL LN AA AC BI GA NA    <219>
453 DATA KG HK KA AA CA BJ LE OA IN    <200>
454 DATA BA JA CE KJ AA IN NB KI HD    <227>
455 DATA CA OJ LE JA CO MJ ME PA PI    <132>
456 DATA FB JJ AA AC MI OI CA BJ NF    <167>
457 DATA LE KJ EF IN NB KI CA AE MM    <58>
458 DATA LF JJ AA AC EM OH LD KA NG    <69>
459 DATA AA CA CA LE CJ HP JJ AA DF    <4>
460 DATA AC OI MI MJ CA PA AH OA HC    <153>
461 DATA BB JA OO EM MO MA II CA BB    <136>
462 DATA BJ LE EM NG LD OI LN AA EH    <127>
463 DATA AC MJ CA PA PI LN AA AC JC    <60>
464 DATA PA AN MJ DL NA AE KJ ME EC    <237>
465 DATA NA AI MJ CC PA AB GA LN NB    <190>
466 DATA AA AC JJ AA AC OI MI MJ BG    <245>
467 DATA AA NA PE JJ AB AC BI JI BA    <108>
468 DATA GJ AE KI GI GI KJ AB IF BE    <63>
469 DATA HL KJ PP IF HK GA LA AD DF    <104>
470 DATA EM ML LD KG HK KA AA EM NG    <91>
471 DATA NG LD CA BM MA CA GP LF MJ    <140>
472 DATA CA NI LA KN HE LH PA DD KD    <51>
473 DATA KN PO AB BI GN HF LH KI AF    <88>
474 DATA KN PP AB GJ AA IF GC IE IB    <31>
```

```
475 DATA GD DI KC JA CA DF MA CA AC      <158>
476 DATA DO MA KC PP OI LN AA AB EF      <207>
477 DATA PA AF JN HH AC NA PF KJ HJ      <206>
478 DATA CA JN HH AC OI OA AG NA NE      <33>
479 DATA PG IG MG CA LE MA IG HK NG      <186>
480 DATA IE HL CA HD AA KK PA LC NO      <193>
481 DATA KC PP IG DK IO MG KI IG OD      <182>
482 DATA DL KC AB IO NC KI JA BF IL      <61>
483 DATA KJ AB IN ML KI EO HE LH CD      <170>
484 DATA DI CA FB LE CA HD AA CA BA      <237>
485 DATA KJ LA EM LN MA CA OB KI ML      <46>
486 DATA KF FA KG FB IF BE IN PO BA      <31>
487 DATA AB IG BF IO PP AB KJ AA ND      <14>
488 DATA IN ML KI EM NN MA KN ML GB      <241>
489 DATA KI NA AL KJ KN IF DO KJ EF      <40>
490 DATA MG IF DN EM JC LD KJ KO HA      <1>
491 DATA IF DO KJ LO IF DN EM JC MK      <182>
492 DATA LD KJ KO IF DO KJ JO IF JJ      <3>
493 DATA DN EM JC LD EM OH MA CA OB      <36>
494 DATA LO LC KF EO IF BE KF EP PA      <205>
495 DATA IF BF CA HK MA JA ON KJ BK      <190>
496 DATA OA KC AA IN EN LG IO EM OM      <43>
497 DATA LG KF FP IF HK IN EO LG EK      <178>
498 DATA KF GA IF HL IN EP LG KJ EA      <227>
499 DATA AA IN MP KI IN MI KI CA CB      <80>
500 DATA CD LD EM FE LF CA GP LF GP      <245>
501 DATA CA NG LN CA LO LF EM IN BP      <190>
502 DATA MA KJ AA IN MK KI IF DL CI      <193>
503 DATA IN MG KI CA HK LF CA HK OE      <120>
504 DATA LF CA HK LF OG DL CA HK LP      <53>
505 DATA LF EM EI LF KJ AO CA MD JI      <134>
506 DATA PP KJ AA IN MI KI GA KJ KO      <203>
507 DATA OA KC AA IF PM IG PL KJ CN      <202>
508 DATA AA IF IL IF IM IN FB LG LF      <13>
509 DATA IN FA LG KA AB IM NC KI DK      <124>
510 DATA IM MH KI KM MI KI NA AK PB      <103>
511 DATA KO MG KI OI CA EH MA CA EL      <184>
512 DATA HB MA CA NJ LF KN FA LG JC      <114>
513 DATA NA AO OO MG KI KN EO LG OL      <249>
514 DATA KO EP LG IF HK IG HL GA BD      <10>
515 DATA EM BE KN KO EM LG KN EN LH      <99>
516 DATA LG CA EH MA KJ CN CA AM NP      <214>
517 DATA OB CA FE LO KF AF KG AE GH      <181>
518 DATA CA EH MA EM HB MA CA EA AE      <28>
519 DATA LG II LB HK PA FK IN MO AO      <19>
520 DATA KI II IE CJ LB HK IN MN GC      <92>
521 DATA KI KA AE CA ND KI CA EL FC      <37>
522 DATA LO CA KB LB KG CJ PA BO AN      <114>
523 DATA DA AP KA AA KF DL PA AJ LI      <49>
524 DATA LJ EN AA JB AE MI MK NA PN      <152>
525 DATA PH KF CJ CJ HP BI GF PL OF      <179>
526 DATA IF PL JA AC OG PM CA NP PD      <186>
527 DATA LP KN MG KI MJ AC NA AD HI      <59>
528 DATA CA JI LP KN FB LG NA BA AL      <248>
529 DATA KN MO KI IF HL KN MN KI EF      <63>
530 DATA IF HK CA FG MA EM NJ LF AP      <14>
531 DATA KJ AA IN FB LG KA AD LB JB      <75>
532 DATA HK IF DK II LB HK IF DJ KK      <32>
533 DATA GA FA MD AB AI AA AA KJ CF      <49>
534 DATA AB IN FB LG EM HD AA CA HE      <192>
535 DATA LO LC KF EO EI KF EP EI OH      <167>
536 DATA CA MD MA CA LO LC CA FE KH      <196>
537 DATA KK GI IF EP GI IF EO KA MB      <209>
538 DATA AA KF FA JB EO KF FB PA LK      <144>
539 DATA AD MI JB EO GA CA LO LC JK      <203>
540 DATA EM FN KK KK LN LN LG EI HF      <14>
541 DATA CA HD AA CA LO LC DI KF AA      <27>
542 DATA EO OF PL KK KF EP OF PM KN      <120>
543 DATA JA AM NA CH MK MK BA BA EH      <111>
544 DATA OA PO JA BP LA AK MK EJ FK      <36>
545 DATA PP NA BI MK PA AC BA BD MG      <101>
546 DATA GI IF EN IG EO KJ AC IF DO      <204>
547 DATA CJ GA JA LA PA DA NA BA MJ      <109>
548 DATA AA FA HA EM AI KN FA FC GD      <70>
549 DATA EP EH FC EB EN BN FD EJ CP      <93>
550 DATA FK EF BN DN CA FG EB FC AC      <100>
551 DATA EJ EB EC EM EF FD BN DN AK      <241>
552 DATA BN BN BN CA EG FC EF EF JJ      <174>
553 DATA BN EN EF EN EP FC FJ BN BD      <29>
554 DATA DN BN CA BN EC FJ FE EF ML      <86>
555 DATA FD CA EL EF FJ CA EI CA OE      <63>
556 DATA BD LH GI CA EH MA KA CN CG      <152>
557 DATA CA BD LH KJ AN EM NC PP LN      <181>
558 DATA LJ MJ LG CA NC PP MI MJ LK      <198>
559 DATA CA NA PF GA KF CN DI OF DE      <75>
560 DATA CL KK KF CO OF CM KI KA AB      <76>
561 DATA AA CA AB LH KF PN DI OF JH      <79>
562 DATA AC KK KF PO OF AD KA AP OG      <76>
563 DATA CA AB LH KF AC DI OF CN MJ      <223>
564 DATA KK KF AD OF CO KA BO EM GP      <118>
565 DATA AB LH KA AH IF DN AG DN GE      <247>
566 DATA KJ DA JA AC KJ DB CA AM HB      <172>
567 DATA OB II BA PC GA KJ CF CA LJ      <209>
568 DATA AM OB KF EP CA EN LH KJ KO      <8>
569 DATA CA CA AM OB KF EO EM EN LJ      <143>
570 DATA LH AA AF CA LO LC KF EO DP      <24>
571 DATA IN PO AB KF EP IN PP AB AN      <139>
572 DATA CA MD MA CA DG LN KF EO KJ      <74>
573 DATA IN HF LH GI GI OO HE LH KC      <167>
574 DATA KM PO AB KN PP AB EM HI BM      <104>
575 DATA LE CA HC LB CA FG MA KA MN      <57>
576 DATA AB LB FP PA GH MI LB FP EA      <182>
577 DATA KK MI LB FP MF BF NA AE DA      <113>
578 DATA OE BE PA AC LA FG EI CA FI      <42>
579 DATA BC LN CA HB MA GI CA EH OP      <155>
580 DATA MA KJ AA IF AP KF GB IN JA      <96>
581 DATA NA KI KC PP OI LN AA AB LP      <127>
582 DATA NA PK IO NA KI KJ CA CA LJ      <160>
583 DATA AM OB OO NA KI KM NA KI HH      <249>
584 DATA MA AG NA PD KA AE LB FP DN      <158>
585 DATA PA CD DA BA CA AM OB OO EO      <189>
586 DATA NA KI MJ CC NA AC IF AP MJ      <184>
587 DATA MI NA OL GA KG AP NA OM FE      <115>
588 DATA MJ ME NA AC IF AP CA CA DD      <154>
589 DATA LB EM PL LH GA KA AA LB GA      <83>
590 DATA FP KK MI LB FP IF GA IG EM      <32>
591 DATA FP EM JP LH KJ AA IF ND AC      <53>
592 DATA IF NG EM BA OF CA BM LK JC      <118>
593 DATA CA BP LI CA BA OF OG NG MI      <103>
594 DATA KF NG MJ BJ LA OG KA AA JD      <244>
595 DATA LB NB MJ DA JA ON MJ DK PL      <89>
596 DATA LA OJ KF NB KG NC IF HK IG      <84>
597 DATA IG HL CA OB KI CA HN LI PP      <45>
598 DATA JA MK KC AA IO KF AC OO BP      <84>
599 DATA JC AC CA IB OJ KF NJ AJ KF      <225>
600 DATA IA IF NJ CA BP LI KJ AA HO      <246>
601 DATA IN NA KI KA AC LB FP IF DM      <199>
602 DATA BE KK MI LB FP IF BF EM HM      <100>
603 DATA MB LH KF CL KG CM IF FP PO      <149>
604 DATA IG GA EI KA AB LB FP PA MP      <96>
605 DATA CL EI II LB FP IF FP GI FH      <107>
606 DATA IF GA KA AD LB FP MF FB KO      <248>
607 DATA PA AM LA BI KF GA KK GI NL      <117>
608 DATA KF FP EI DI LA NN II LB EK      <118>
609 DATA FP MF FA NA OP DI GI IF FI      <57>
610 DATA FP IG GA GA GI BI GA KJ CO      <112>
611 DATA AA IF ND KJ BI IF NG EM MA      <25>
612 DATA BA OF CA BM LK CA LK LI HN      <138>
613 DATA MG NG CA BA OF MG NG DA HN      <205>
614 DATA OG KA AA LB NB MJ DA JA JB      <160>
615 DATA PB MJ DK LA ON KF NB KG KN      <97>
616 DATA NC IF HK IG HL CA OB KI HL      <74>
617 DATA KF FA IF BE KF FB IF BF BO      <193>
618 DATA CA HK MA JA MC CA LK LI DO      <28>
619 DATA KA AA IM NA KI LB FP KK FO      <15>
620 DATA MI LB FP IF GA IG FP LB FD      <168>
621 DATA FP PA KM CA GJ LI EM HB PJ      <179>
622 DATA MA EM MO MA KA AA LB HK GF      <80>
623 DATA MJ CC NA PF MI LB HK PA JD      <77>
624 DATA AM MJ CC PA AF JJ JF AD BN      <34>
625 DATA NA PC CA HG LC CA IE KL FJ      <117>
626 DATA KJ AA JJ JG AD GA CA LO BJ      <246>
627 DATA LC CA MD MA KF EO PA CH FP      <55>
628 DATA KK OA AH LA CC MK IK AK MB      <192>
629 DATA AK AK AK KK CA BH LJ JI FA      <69>
630 DATA PA BC MA AO LA AO KA PP CN      <164>
```

```
631 DATA MK OI MI LJ JG AD JN BA HJ      <245>
632 DATA MB NA PG GA EM MO MA EM AN      <154>
633 DATA CP MA KF MM NA AH KN IN HB      <161>
634 DATA AC MJ AF PA HG KF ML MJ GP      <106>
635 DATA AD PA DP JA DK MJ AH LA HM      <17>
636 DATA DG MF MF PA DC IF MF DI GE      <122>
637 DATA OJ AE AK KO IN AC PA AC CG      <225>
638 DATA GJ AB AK AK AK AK KK KA NM      <76>
639 DATA PP MK OI MI LN BA MB PA PH      <37>
640 DATA BB MJ FP NA AC KJ AN JJ FK      <90>
641 DATA HH AC PA AG MA AO JA OK LH      <65>
642 DATA NA OI IE MG EM EC OL EM MH      <40>
643 DATA EI OL KF MM NA PJ CA AP JM      <11>
644 DATA MA KC AA IG MP OI IG MN PC      <142>
645 DATA OM IN AC NA AJ CA CI LI FE      <11>
646 DATA CA BP LI EM NM LJ CA MF LN      <78>
647 DATA LI CA CO LK CA GM OF KN NO      <215>
648 DATA AO NM AJ AB HI IN AO NM OD      <54>
649 DATA EM EC OL MN IO AC PA CG OM      <3>
650 DATA IN IO AC CA BM LK KC BA MF      <76>
651 DATA KF ND MJ BA JA AC KC AF IK      <195>
652 DATA IG ND KJ AB IF MG CA CO JM      <12>
653 DATA LK KJ BN IF ML IN HH AC NG      <169>
654 DATA CA AF OL CA CO LK EM EI KM      <222>
655 DATA OL CA BA OF KE ND KF MO OK      <139>
656 DATA JB NB CA CE OK KN IH AC MG      <0>
657 DATA JB PD GA CA BA OF KE ND HA      <227>
658 DATA LB NB IF MO CA CE OK LB LE      <36>
659 DATA PD IN IH AC KJ AC IF MN AG      <187>
660 DATA GA CA HD AA KC CA IG EN II      <10>
661 DATA CA LO LC KA AD IE CJ GA EA      <45>
662 DATA KC EM CA HD AA MJ CI NA EC      <180>
663 DATA ON CA HD AA CA LO LC KJ LJ      <83>
664 DATA NC CA MF MA KC GM IG EN FI      <68>
665 DATA NA OB KA PP MI LB HK NA BD      <33>
666 DATA PL MI CA ND KI KA AA GA FO      <58>
667 DATA CA HG LC CA HF KM KN MG PM      <179>
668 DATA KI NA AC LA EO CA PA KM DE      <126>
669 DATA KA AC KF HK JB DP MI KF PO      <167>
670 DATA HL JB DP CA GN LK KA AD DF      <40>
671 DATA MI LB HK PA AK BA PJ MJ LP      <121>
672 DATA LI PA BE MJ LJ PA CA KA OO      <166>
673 DATA AA LB HK KK MI LB HK PA LI      <21>
674 DATA CF IG HK IF HL NA NP KA HE      <174>
675 DATA AA LB HK IN MN KI MI LB KG      <105>
676 DATA HK IN MO KI EM EB LN CA OH      <220>
677 DATA IE KL CA HL LK CA HD AA BH      <99>
678 DATA EM KK LK EM AO KN EM BE BH      <84>
679 DATA KN EM BB KN CA HG LC CA BP      <63>
680 DATA HI KL CA DK KL JA OM LK FO      <172>
681 DATA OA CI JA ON KN MN KI EI OP      <163>
682 DATA KN MO KI EI KN NC KI EI NK      <214>
683 DATA OO FA LG OO MH KI KN MH MF      <67>
684 DATA KI PA NG IN NC KI KA AA BF      <56>
685 DATA IE AJ IE FP LB HK KK KA OF      <223>
686 DATA AC LB DP EI MI LB DP OA NC      <166>
687 DATA ND NA DB EI CA CF LO GI IH      <223>
688 DATA IF HL GI IF HK CA MD MA AK      <140>
689 DATA KG FP LN JG AD IF EO OI BG      <139>
690 DATA LN JG AD IF EP OI IG FP PH      <190>
691 DATA CA HF KM CA NC KM CA KD KC      <131>
692 DATA KM CA BF LB PA AL CA MD HA      <178>
693 DATA MA EM CL LL IF HL GI IF NP      <193>
694 DATA HK KF AJ MF FP NA BL CA FH      <150>
695 DATA BF LB NA BG CA GN LK CA BD      <167>
696 DATA NJ LF MO FA LG GI IN NC CJ      <218>
697 DATA KI GI IN MO KI GI IN MN NF      <149>
698 DATA KI GA EM BK KN CA LO LC KL      <28>
699 DATA KF EO KG EP IF BE IG BF BM      <175>
700 DATA CA HK MA CA MD MA CA LO NL      <134>
701 DATA LC KF EO KG EP IF BE IG LJ      <171>
702 DATA BF CA MD MA CA DG LN KA GL      <214>
703 DATA AC KF BE JB FP BI GF EO HG      <3>
704 DATA IF BE KF BF MI JB FP GJ HE      <64>
705 DATA AA IF BF LA BC KA AA LB KN      <1>
706 DATA FP KK MI LB FP PA AH IF FN      <208>
707 DATA GA IG FP MI NA NL GA EM GE      <131>
708 DATA CP MA KA AB JI JB CL CA AE      <68>
709 DATA JP MA KF CC KG CD BI GJ HA      <25>
710 DATA AC JA AB OI IF CN IG CO OB      <96>
711 DATA EM IN MA CA LE MA KJ AA NG      <199>
712 DATA IN ML KI CA FB LE KA PP ME      <0>
713 DATA KC PP KN AA AC PA FC MJ FL      <53>
714 DATA CC NA AB MI MI OI LJ AA CE      <170>
715 DATA AC JN DM AD NA PG IG CI FC      <175>
716 DATA KF CL IF FP KF CM IF GA GK      <70>
717 DATA KA AE IE AJ KC AA KE AJ IA      <127>
718 DATA LB FP PA BP LN DM AD MJ OE      <60>
719 DATA DP PA AI MJ CK PA CD NB AO      <19>
720 DATA FP NA AM OI OE CI PA BK DJ      <110>
721 DATA MI LB FP PA AG NA OF OG GJ      <253>
722 DATA AJ NA NJ KA AA LB FP KK AM      <102>
723 DATA MI LB FP IF GA IG FP NA HC      <47>
724 DATA MH GA KA AC LB FP IF BE HC      <20>
725 DATA MI LB FP IF BF CA KC LH OL      <163>
726 DATA KA AB LB FP PA OL EM AL OD      <18>
727 DATA LM KF ML MJ EA PA PK GA HP      <89>
728 DATA CA EO LP KJ AA IF JA KN JI      <214>
729 DATA MM KI CA LB PP CA AF MB CK      <165>
730 DATA KJ PA CA JD PP KJ CE CA DI      <164>
731 DATA KI PP CA KO PP KN MM KI JF      <203>
732 DATA CA LE PP KJ GA CA JG PP JB      <214>
733 DATA KC AF CM KC AD CA LB LM AF      <17>
734 DATA MK NA PK KK CA HB MA CA KP      <34>
735 DATA BC LN CA LB LM CA EH MA ID      <119>
736 DATA CA AN LN CA LB LM KK PA BB      <36>
737 DATA OC CA AM OB NA PF CA KF HJ      <11>
738 DATA PP KE JA NA AB GA GI GI DE      <18>
739 DATA CA KL PP KN MM KI CA LB LM      <179>
740 DATA PP KJ OA CA JD PP EM KO DE      <88>
741 DATA PP CA DG LN KF EO IN MM FO      <103>
742 DATA KI GA KC AA IG DN OI KA PF      <158>
743 DATA DE CA BD LH KJ AA CA EH CO      <253>
744 DATA MA KJ CM CA AM OB KJ CC GN      <56>
745 DATA CA AM OB KF DN AK AK AK AN      <233>
746 DATA AK KK LN BA MB PA AH OI CB      <200>
747 DATA CA AM OB EM PF LM OG DN CN      <199>
748 DATA CA HB MA KG DN OA AG NA OK      <154>
749 DATA ML GA KJ CA EM AM OB KF NC      <107>
750 DATA ML MJ DM PA PK GA CA CF FP      <56>
751 DATA LO KF AJ KK EK KI AJ IA JB      <13>
752 DATA IF CJ II MK LN JG AD NA CG      <50>
753 DATA BC MK LN JG AD JB AE II EP      <59>
754 DATA BA PB GA CA LO LC KF EP OF      <2>
755 DATA NA AB GA EM CP MA KF HK IL      <223>
756 DATA NA AC MG HL MG HK GA CA ND      <206>
757 DATA HD AA CA BH LJ IE AJ II HI      <235>
758 DATA EM MM LO CA LO LC KF EO FJ      <116>
759 DATA EI KF EP EI CA MD MA CA EH      <35>
760 DATA PM MA GI KI GI KK KJ AA IH      <150>
761 DATA CA NF PP CA AF MB CA JP JJ      <107>
762 DATA MA EM NG LN CA LO LC CA EP      <176>
763 DATA MD MA KF EO EI KF EP EI PK      <177>
764 DATA CA LO LC KF EO EI KF EP LP      <20>
765 DATA EI CA MD MA CA PM MA GI CP      <153>
766 DATA KI GI KK GI IF DO GI IF NC      <248>
767 DATA DN CA AP MA KJ DN CA NI AK      <69>
768 DATA PP CA BM MA EM NG LN CA PK      <111>
769 DATA PM MA KF CN DI KE CO OJ IB      <6>
770 DATA AC LA AB II KK KJ AA CA KO      <227>
771 DATA NF PP EM MF LL KJ AA IF MO      <186>
772 DATA JA KJ AB KO MM KI KA GP GL      <199>
773 DATA CA LK PP KJ AA CA LN PP FO      <22>
774 DATA EM MA PP CA EO LP CA MA BI      <71>
775 DATA LN CA HB MA KN MM KI CA EP      <8>
776 DATA LE LE PP KJ GP CA JG PP CA KA   <187>
777 DATA KF PP CA AM OB CE JA FA LF      <132>
778 DATA PG KN MM KI CA KL PP KJ IK      <175>
779 DATA AB EM MD PP CA BH LJ CA BP      <36>
780 DATA EO LP CA LB PP KJ GP CA BF      <147>
781 DATA JD PP KA AA LJ JG AD PA HE      <224>
782 DATA AG CA KI PP MI NA PF KN AH      <169>
783 DATA MM KI CA KO PP KJ AB EM DH      <246>
784 DATA MD PP KJ AA IF AJ CA HG IP      <81>
785 DATA LC CA LO LC KE AJ KF EO OC      <242>
786 DATA JJ JG AD KF EP JJ JH AD FJ      <99>
```

```
787 DATA MI MI IE AJ CA BF LB PA PD      <252>
788 DATA AG CA MD MA EM CM LO GA DP      <237>
789 DATA KN MI KI NA BC KF DL PA MP      <148>
790 DATA AO KF PL DI OF IL IF AE NP      <195>
791 DATA KF PM OF IM IF AF GA KJ KF      <192>
792 DATA DM KC AD IF AE IG AF GA FF      <183>
793 DATA CA HJ AA CA ON MA CA HJ PP      <244>
794 DATA AA MJ MG NA EA KC AB KA OC      <95>
795 DATA AA CA LL LO CA HK MA KF JI      <204>
796 DATA GA KG FP IF DO IG DN CA AL      <41>
797 DATA HD AA CA ON MA KA PP CA PP      <150>
798 DATA LL LO OG BE NA AC OG BF EA      <91>
799 DATA CA HK MA KF CO DI OF GA KK      <212>
800 DATA KK OI KA AA LB FP JB DN BA      <149>
801 DATA MI NA PJ OG DO OG GA MK MF      <234>
802 DATA NA PC EM MK LL EM MO MA GN      <123>
803 DATA KF BE AF BF NA AE IG BE EB      <104>
804 DATA IE BF GA CA CF LO KE AJ KJ      <89>
805 DATA II LJ JG AD JB AE II BA AH      <186>
806 DATA PI KF AJ AJ IA IF CJ GA DN      <209>
807 DATA CA LO LC KA AA LB EO IF LE      <66>
808 DATA EO IE EP EM FN KK CA LO FC      <19>
809 DATA LC KA AA LB EO KK MI LB HE      <184>
810 DATA EO IF EP IG EO EM FN KK EJ      <227>
811 DATA CA HD AA CA LO LC KF EO BG      <28>
812 DATA KG EP IF BE IG BF CA HK MD      <213>
813 DATA MA JA CH KF FP IN MN KI HN      <142>
814 DATA KF GA IN MO KI GA CA DG LO      <219>
815 DATA LN CA MD MA KJ AA IF EP NN      <86>
816 DATA KG EO OA AE JA AJ CA ME FF      <79>
817 DATA LP CA JM LH EM GP LF EM OO      <206>
818 DATA CP MA EM OH MA KC PP OI GL      <147>
819 DATA LN EG LP JN JG AD NA PH LP      <76>
820 DATA EM AC LO EN CN FH AH BM AA      <165>
821 DATA AB AP AA KN MM KI MJ AI AC      <132>
822 DATA LA AC GI GI GA KN MK KI AB      <131>
823 DATA NA DI KN MN KI IN MJ KI CI      <76>
824 DATA KN MO KI IN MK KI EM HG OE      <101>
825 DATA LC KN MK KI PA CE KO MG FJ      <210>
826 DATA KI PA IJ CA HG LC CA LO EH      <81>
827 DATA LC KF EO AF EP PA AG KJ JI      <16>
828 DATA AA IN MK KI GA KN MK KI HO      <145>
829 DATA KO MJ KI IN MO KI IO MN HN      <202>
830 DATA KI GA EM BN KN CA BF LB AE      <95>
831 DATA PA LK EM BH KN CA HG LC AC      <176>
832 DATA CA BH LJ KF DL PA KN CA IN      <45>
833 DATA EO LP KJ AB IN MI KI KO GC      <236>
834 DATA MM KI CA ME LP KN EM LG MG      <97>
835 DATA CA AM OB KN EN LG EM AM BF      <216>
836 DATA OB KA AB KJ AO CA LK PP BC      <191>
837 DATA KF CI KC JG KA AD CA LN IF      <6>
838 DATA PP CA MA PP CA AF MB KC GG      <15>
839 DATA AO EM MJ PP KN MI KI PA CP      <230>
840 DATA BL KF DL PA BH KC AO CA NC      <241>
841 DATA MJ PP KA AA KF CJ CJ HP NO      <240>
842 DATA PA AK KK LB AE CA AM OB GG      <199>
843 DATA MI MK NA PH GA EM FA MA BF      <228>
844 DATA EI KJ DH IF AB GI GA KJ BP      <107>
845 DATA DG IF AB GA KN AO NM CJ NM      <242>
846 DATA PO HI IN AO NM KN AN NM ID      <181>
847 DATA GA KN AO NM AJ AB IN AO JM      <148>
848 DATA NM KN AO NM FI GA CA AD EO      <35>
849 DATA MA EM BM MA CA CJ MA EM DN      <116>
850 DATA EI LC CA AD MA CA EJ LM AC      <67>
851 DATA EM AK MA CA AD MA CA NP PI      <206>
852 DATA LN EM AK MA CA AD MA CA NG      <101>
853 DATA MN LN EM AK MA CA AK MA IK      <126>
854 DATA EM LG LA CA AD MA CA LM HB      <123>
855 DATA PG CA CM KI EM AK MA KF KF      <232>
856 DATA AB EI CA AK MA CA GN LJ HJ      <149>
857 DATA GI IF AB EM EC OL CA AD IK      <222>
858 DATA MA CA NH KK EM AK MA CA JH      <67>
859 DATA AD MA CA BD KG EM AK MA LC      <250>
860 DATA GI KI GI KC PK JK EI JI IO      <101>
861 DATA EI GA CA ID MA EM LN MA NE      <22>
862 DATA CA AD MA CA EE KG CA AK BH      <195>
863 DATA MA EM MI LA CA AD MA CA IH      <54>
864 DATA DD KF EM AK MA CA CJ MA PH      <127>
865 DATA EM DF KE CA AK MA EM FN LI      <18>
866 DATA LE CA CJ MA CA GA KF EM CO      <39>
867 DATA AK MA CA CJ MA EM HE KE DH      <34>
868 DATA KJ ND KA AA NB HK NA AD DK      <213>
869 DATA EM HD AA CA CJ MA EM AI BM      <192>
870 DATA KP CA CJ MA CA MM PP EM OP      <245>
871 DATA GC KE BI CA FB LE CA AD GG      <168>
872 DATA MA EM KC KE CA CJ MA EM KH      <155>
873 DATA OD KI CA AD MA CA HJ AA AH      <134>
874 DATA CA GL KJ CA AK MA EM HJ OD      <243>
875 DATA AA CA AD MA CA NE OB EM AE      <122>
876 DATA AK MA JA AI EI CA CJ MA LD      <181>
877 DATA GI EM PJ OA GA GA PP PP EL      <168>
2000 DATA-1                              <71>
  ENDE DES LISTINGS
   [DRUCK:F.B.]
```

CPU - Bewerbung als Programmautor - CPU

an den: Tronic-Verlag, Abteilung Programmentwicklung, Landstraße 29, 3444 Wehretal 1

Name: _____ Vorname: _____ Alter: _____

Straße: _____ Wohnort: _____

Titel des Programmes: _____ Computersystem: _____

Erforderlicher Speicherplatz: _____ kBytes

Erforderliche Pheriphere: _____

Ich versichere hiermit, daß ich der Autor des oben genannten Programmes bin und alle Rechte besitze. Ich bin damit einverstanden, daß Sie mein Programm in einer Ihrer Zeitschriften abdrucken und / oder dieses andersweitig kommerziell verwenden. Das einmalige Honorar beträgt im Falle einer Veröffentlichung 120,- DM pro komplett abgedruckte Seite. Sollte das Programm nicht veröffentlicht werden, so bitte ich um Rücksendung meiner Unterlagen.

Folgende Unterlagen liegen diesem Schreiben bei:

Kassette: _____ Diskette: _____ Dokumentation: _____ Listing: _____ Rückporto: _____

Unterschrift des Programmautors: _____

SPECTRUM

Übersetzung von Grafik in Maschinencode

ZX-Spectrum: Grafik-Compiler

SPECTRUM

Dieses Programm ist ein Compiler, das Grafik in Maschinencode übersetzt.
Startet das Programm, fragt es zuerst nach dem RAMTOP! Will man das Maschinenprogramm ab 60000 ablegen, dann gibt man 59999 ein.
Dann fragt das Programm nach der Zeilenlänge bzw. Zeilenanzahl: Bei der Zeilenanzahlfrage gibt man zum Beispiel 100 ein. (Man kann dann ein Programm eintippen, das 100 Zeilen lang ist!!!) Wenn nach der Zeilenlänge gefragt wird, muß man eine Zahl eingeben, die groß genug ist, da man sonst z. B. den PRINT-Befehl in zwei Zeilen schreiben muß, was später mehr Platz im MC einnimmt. (Am besten gibt man 45 ein, denn dann kann ein PRINT-Befehl 32 Zeichen enthalten.)

Anschließend erscheint das MENÜ:
1 EINGABE
2 ÄNDERN
3 AUSGABE
4 LÖSCHEN
5 ÜBERSETZEN
6 SAVE
7 LOAD
8 EINFÜGEN

Folgende Befehle kennt der Compiler:

(WSB = Wie Sinclair BASIC ; _ = Hier muß ein Leerraum stehen!)

POKE_x,y	: WSB
PEEK_x	: PRINT PEEK x
PLOT_x,y	: WSB
DRAW_x,y	: WSB
CLS	: WSB
INIT_x	: Ausgabegerät (0/1 = Unterer Bildschirm; 2 = Bildschirm; 3 = Drucker)
AT_x,y	: PRINT AT x,y;
PAUSE_x	: WSB
PRINT$_x$	: Gibt einen String aus; Wichtig: Am Ende des Strings, muß ein " stehen, welches das Ende des Strings anzeigt.
PRINT#x	: Gibt eine Zahl aus; Zahl im Bereich von 0–9999
PRINT%_x$	: Gibt ein Zeichen (x$) aus.
INK_x	: WSB
PAPER_x	: WSB
BORDER_x	: WSB
COPY	: COPY, aber 192 Zeilen!!!
OVER1	: WSB
OVERO	: WSB
INVERSE1	: WSB
INVERSO	: WSB
HOME	: PRINT AT 0,0;
SCROLL	: Scrollt eine Zeile nach oben
CIRCLE_x,y,z	: WSB
FLASH1	: WSB
FLASHO	: WSB
BRIGHT1	: WSB
BRIGHTO	: WSB
FLASH8	: WSB
BRIGHT8	: WSB
BREAK	: Wartet bis man BREAK drückt.
KEY_X$_	: Wartet bis man die Taste x$ drückt.
DI	: Interrupt sperren.
EI	: Interrupt freigeben.
REM X$	: WSB

ENDE beendet die Eingabe!!!

Anmerkung von Michael Sassin; Autor des Monitorprogramms für ATARI 400/800 XL:

Durch die Inkompatibilität der Operationssysteme der alten 400/800- und der XL-Reihe tritt das Problem auf, daß der Befehl 'P' bei den XL-Computern nicht zum gewünschten Umschalten auf den Drucker führt. Es wird im OS eine Endlosschleife angesprungen, die nun nicht wieder verlassen kann. Um das Problem zu lösen, lädt man entweder das alte OS, bevor das Monitorprogramm geladen wird oder man verbessert die jeweiligen Zeilen in den Basic-Programmen (Disketten-/Cassettenversion).

Monitor/Diskettenversion

430 DATA 30208B20606936208B20602093209830B560EAAAAD470348AD4603488AA2
440 DATA 0060A001A201201420A000E82014206CB20020BD25203621A90085B8A5B3

Monitor/Cassettenversion

440 DATA 6936208B08602093089830B560EAAAAD470348AD4603488AA20060A001A2

zu 2) Hiermit können Zeilen abgeändert werden.
Drückt man W wird die nächste Zeile gelistet.
Drückt man A kann man die Zeile neu eingeben.
Drückt man E kehrt man ins MENÜ zurück.

zu 3) Hiermit kann man das Programmlisting ausgeben.

zu 4) Hiermit können Zeilen gelöscht werden (sonst wie 2).

zu 5) Hiermit wird das Programm übersetzt. Tritt ein Fehler auf, wird er mit der Zeile angezeigt (Invers), drückt man dann eine Taste, kehrt man zum MENÜ zurück. Hat der Compiler keine Fehler gefunden, so wird angezeigt, wie man den Code SAVEn kann.

zu 6) Hiermit wird das Programmlisting auf Cassette abgespeichert. Name max. 9 Buchstaben!

zu 7) Hiermit wird das Programmlisting von Cassette geladen. Es muß ein Name angegeben werden! (max. 9 Buchstaben)!

zu 8) Will man eine Zeile freimachen, so gibt man die Zeilennummer deren ein, die freigemacht werden soll. Alle darauffolgenden Zeilen werden um eins nach oben verschoben. Danach kehrt man zum MENÜ zurück.

Wichtig!!!
Es darf nur ein Befehl pro Zeile vorkommen. Nach den meisten Befehlen muß ein Leerraum stehen (siehe 1). Es dürfen keine TOKEN benutzt werden. Der Befehl KEY/BREAK funktioniert auch dann, wenn der Interrupt gesperrt ist. Der Interrupt muß am Ende des Programms wieder freigegeben werden.

Zeile:		Bemerkung:
1-	6	Der RAMTOP wird erfragt und gesetzt. Will man den Code ab 60000 abgelegt haben, so gibt man 59999 ein.
	7	Umschaltung auf Großbuchstaben.
10-	70	Befehle werden in das Stringarray b$ eingelesen.
	80	Dimensionierung Programmspeicher.
100-	210	Menü
300-	350	Eingabe
400-	435	Ändern
500-	508	Ausgeben
600-	635	Löschen
700-	725	Übersetzen

701: Befehle EXX:PUSH HL
702-713: Befehlsauswertung und Sprung zum entsprechenden Unterprogramm
720: Nach Bearbeitung UP, Rücksprung nach 720
721-725: Befehle: POP HL:EXX:RET

800- 812	Save
900- 910	Lead
1000-1006	Einfügen

Unterprogramme:

1500-1600	poke
1750-1800	peek
2000-2056	plot
2250-2400	draw
2500-2600	cls
2750-2800	init
3000-3100	at
3250-3300	pause
3500-3604	print$
3750-3800	print#
4000-4100	print%
4250-4300	ink
4500-4600	paper
4750-4800	border
5000-5100	copy
5250-5300	over1
5500-5600	over0
5750-5800	inverse1
6000-6100	inverse0
6250-6300	home
6500-6600	scroll
6750-6800	circle
7000-7100	flash1
7250-7300	flash0
7500-7600	bright1
7750-7800	bright0
8000-8100	flash8
8250-8300	bright8
8500-8600	break
8750-8800	key
8900-8901	di
8910-8911	ei
9000-9100	Fehlermeldungen.
9999	Save-Routine.

IMPRESSUM

Tips & Tricks von CPU
erscheint alle 2 Monate im
Tronic-Verlag, 3444 Wehretal 1
Tel. 05651/40693 oder 40643

Herausgeber:
Axel Credé
Chefredakteur:
Siegfried Görk
Redaktion:
Manfred Kleimann, Hartmut Wendt, Frank Brall, Ottfried Schmidt.

Programmautoren:
W. Sohrt, R. Lässig, H. Kusch, F. Brall, M. Sassin, U. Schröder, H.-J. Köhnen, G. Raskopp, W. Bönigk, T. Fippel, P. Schmitz, J. Otto, D. Twigg-Flesner, W. Raaz, K. Rupprecht u. a.

Gesamtherstellung:
Druckhaus Dierichs Kassel, Frankfurter Str. 168, 3500 Kassel

Vertrieb:
Inland (Groß-, Einzel- und Bahnhofsbuchhandel) sowie Österreich und Schweiz:
Verlagsunion
Friedrich-Bergius-Straße 20
6200 Wiesbaden
Telefon 06121/2660

Anzeigenverwaltung:
Tronic-Verlag GmbH
Landstraße 29
3444 Wehretal
Telefon 05651/40643 oder 40693
Telefax 05651/40685

Anzeigenleitung:
Inland: H. Wendt
Ausland: M. Kleimann

Erscheinungsweise:
Erstverkaufstag von
Tips & Tricks von CPU ist Anfang des Monats

Anzeigenpreisliste:
Bitte Mediaunterlagen anfordern.

Software-Service und Vertrieb:
Annelie Kratzenberg, Heike Rabe

Urheberrecht:
Alle in HOMECOMPUTER veröffentlichten Beiträge sind urheberrechtlich geschützt. Alle Rechte, auch Übersetzungen, vorbehalten.
Reproduktionen jeder Art (Fotokopien, Microfilm, Erfassung in Datenverarbeitungsanlagen usw.) bedürfen der schriftlichen Genehmigung des Verlags. Alle veröffentlichte Software wurde von Mitarbeitern des Verlages oder von freien Mitarbeitern erstellt.
Aus Ihrer Veröffentlichung kann nicht geschlossen werden, daß die beschriebenen Lösungen oder Bezeichnungen frei von Schutzrechten sind.

Bezugspreise:
Einzelheft 6,— DM,
Abonnements Ausland:
Tips & Tricks von CPU (6 Hefte):
Europäisches Ausland 43,— DM
Außereuropäisches Ausland 60,— DM
Tips & Tricks von CPU (6 Hefte) & COMPUTRONIC (6 Hefte):
Europäisches Ausland 87,50 DM
Außereuropäisches Ausland 120,— DM
Super-Kombi:
HOMECOMPUTER (6 Hefte) & COMPUTRONIC (6 Hefte) & TIPS & TRICKS von CPU:
Europäisches Ausland 127,50 DM
Außereuropäisches Ausland 170,— DM

Autoren, Manuskripte:
Der Verlag nimmt Manuskripte und Software zur Veröffentlichung gerne entgegen. Sollte keine andere Vereinbarung getroffen sein, so gehen wir davon aus, daß Sie mit einem Honorar von 120,–DM pro abgedruckter Seite im Heft einverstanden sind.

Titelblatt:
Axel Credé, Eschwege

DEMO-Programme

```
CLS                                  Bildschirm löschen
INIT 2                               Ausgabe auf Bildschirm
HOME
PLOT 0,0
DRAW 255,0
DRAW 0,175
DRAW -255,0
DRAW 0,-175
DRAW 255,175
PLOT 255,0
DRAW -255,175
CIRCLE 127,87,80
INIT 2                               Ausgabe auf Bildschirm
AT 1,0                               PRINT AT 1,0;
FLASH1                               FLASH 1
PRINT$ TEST DER BEFEHLE : "PRINT "TEST DER
AT 2,0                               BEFEHLE : "
FLASH0
PRINT$ CLS/INIT/HOME/PLOT/DRAW/AT"
AT 3,0
PRINT$ CIRCLE/PRINT$/FLASHO/KEY"
AT 5,0
PRINT$ G DRUECKEN ! "
KEY G                                Warten bis G gedrückt wird
ENDE                                 ENDE

CLS
INIT 2
HOME
PAPER 0
PRINT$ _____ " :_=32 Spaces
PAPER 1
PRINT$ _____ "
PAPER 2
PRINT$ _____ "
BRIGHT 1
PRINT$ _____ "
PAPER 1
PRINT$ _____ "
PAPER Q
PRINT$ _____ "
BRIGHT0
PAPER 1
PRINT$ TEST DER FARBBEFEHLE ! "
AT 21,0
PRINT$ BITTE A DRUECKEN ! "
KEY A
ENDE
```

```
0>REM     © by S.Slabihoud
                in 1985
          Gustav-Ohm-Str.72
            4250 Bottrop
          Tel.:02041/23488
```

```
  1 CLS
  2 INPUT "Ramtop : ";ram: IF R
AM<23552 THEN LET ADR=RAM: GO TO
 7
  3 POKE 23728,ram-256*INT (ram
/256): POKE 23729,INT (ram/256)
  4 CLEAR ram
  5 LET adr=(PEEK 23728+256*PEE
K 23729)+1
  6 LET RAM=ADR
  7 POKE 23658,8
 10 LET an=33: INPUT "MAX. ZEIL
ENLAENGE (MIN.16) ";LEN: INPUT "
MAX. ANZAHL ZEILEN : ";INS
 20 DIM b$(an,10): REM  Befehle
 30 DIM b(AN):      REM  Zeilen#
 40 DEF FN a(x)=x-256*INT (x/25
6): DEF FN b(x)=INT (x/256)
 50 FOR z=1 TO an
 60 READ b$(z),b(z): NEXT z
 70 DATA "POKE",1500,"PEEK",175
0,"EI",8910,"DI",8900,"KEY",8750
,"BREAK",8500,"REM",720,"FLASH1"
,7000,"FLASH0",7250,"BRIGHT1",75
00,"BRIGHT0",7750,"FLASH8",8000,
"BRIGHT8",8250,"CIRCLE",6750,"HO
ME",6250,"PLOT",2000,"DRAW",2250
,"CLS",2500,"INIT",2750,"AT",300
0,"PAUSE",3250,"PRINT$",3500,"PR
INT#",3750,"PRINT%",4000,"INK",4
250,"PAPER",4500,"BORDER",4750,"
COPY",5000,"OVER1",5250,"OVER0",
5500,"INVERSE1",5750,"INVERSE0",
6000,"SCROLL",6500
 80 DIM P$(INS,LEN): LET P=0
100 CLS : LET ADR=RAM
110 PRINT '" 1    EINGABE"
120 PRINT '" 2    AENDERN"
130 PRINT '" 3    AUSGEBEN"
140 PRINT '" 4    LOESCHEN"
150 PRINT '" 5    UEBERSETZE
N"
160 PRINT '" 6    SAVE"
170 PRINT '" 7    LOAD"
180 PRINT '" 8    EINFUEGEN"
199 FOR A=1 TO 16 STEP 2: OVER
1: INK 8: PAPER 8: BRIGHT 8: PRI
NT AT A,0; BRIGHT 1;"
          ": NEXT A: OVER 0: INK
 0: PAPER 7: BRIGHT 0
200 LET I$=INKEY$: IF I$<"1" OR
 I$>"8" THEN GO TO 200
210 GO TO 200+(VAL I$*100)
```

```
 300 REM EINGABE
 301 CLS
 302 FOR Z=1 TO P: PRINT (" " AN
D Z<100);(" " AND Z<10);Z;" ";P$
(Z): NEXT Z
 303 LET P=P+1: IF P>INS THEN LE
T P=P-1: GO TO 350
 304 INPUT ((" " AND P<100);(" "
 AND P<10);P;" "); LINE A$
 305 IF LEN A$>LEN THEN GO TO 30
4
 306 LET P$(P)=A$: IF A$="ENDE"
THEN LET P=P-1: GO TO 100
 307 PRINT (" " AND P<100);(" "
AND P<10);P;" ";P$(P)
 308 GO TO 303
 350 PRINT #0;AT 1,0;" Speicher
voll ! ": PAUSE 1: PAUSE 0: GO T
O 100
 400 REM AENDERN
 401 CLS
 402 LET Z=0
 403 LET Z=Z+1: IF Z>P THEN GO T
O 421
 404 INPUT INKEY$: PRINT (" " AN
D Z<100);(" " AND Z<10);Z;" ";P$
(Z)
 405 PRINT #0;AT 1,0;"E)NDE  W)E
ITER  A)ENDERN "
 406 LET I$=INKEY$: IF I$<>"" TH
EN GO TO 406
 407 LET I$=INKEY$: IF I$="" THE
N GO TO 407
 408 LET I$=INKEY$
 409 IF I$="E" THEN GO TO 100
 410 IF I$="A" THEN GO TO 430
 420 GO TO 403
 421 PRINT "    ENDE"
 422 PAUSE 1: PAUSE 0: GO TO 100
 430 PRINT AT 21,0;: INPUT INKEY
$: PRINT : PRINT : INVERSE 1: PR
INT (" " AND Z<100);(" " AND Z<1
0);Z;" ";P$(Z)
 431 INVERSE 0
 432 INPUT ((" " AND Z<100);(" "
 AND Z<10);Z;" "); LINE A$
 433 IF LEN A$>LEN THEN GO TO 43
2
 434 LET P$(Z)=A$+"

                    "
 435 PRINT AT 21,0;: INPUT INKEY
$: PRINT : PRINT : LET Z=Z-1: GO
 TO 403
 500 REM AUSGEBEN
 501 CLS
 502 INPUT "PRINTER ? (J/N) ";N$
: IF N$="J" THEN OPEN #2,"P"
 503 FOR Z=1 TO P
 504 PRINT (" " AND Z<100);(" "
AND Z<10);Z;" ";P$(Z): IF INKEY$
="S" THEN GO TO 100
 505 NEXT Z
 506 PRINT "    ENDE"
 507 CLOSE #2
 508 PAUSE 1: PAUSE 0: GO TO 100
 600 REM LOESCHEN
 601 CLS
 602 LET Z=0
 603 LET Z=Z+1: IF Z>P THEN GO T
O 621
 604 INPUT INKEY$: PRINT (" " AN
D Z<100);(" " AND Z<10);Z;" ";P$
(Z)
 605 PRINT #0;AT 1,0;"E)NDE  W)E
ITER  L)OESCHEN "
 606 LET I$=INKEY$: IF I$<>"" TH
EN GO TO 606
 607 LET I$=INKEY$: IF I$="" THE
N GO TO 607
 608 LET I$=INKEY$
 609 IF I$="E" THEN GO TO 100
 610 IF I$="L" THEN GO TO 630
 620 GO TO 603
 621 PRINT "    ENDE"
 622 PAUSE 1: PAUSE 0: GO TO 100
 630 PRINT AT 21,0;: INPUT INKEY
$: PRINT : PRINT : IF Z=P THEN L
ET P=P-1: LET P$(Z)="

                    ": GO TO 10
0
 631 FOR C=Z TO P-1
 632 LET P$(C)=P$(C+1)
 633 NEXT C: LET P=P-1
 634 PRINT "ZEILE ";Z;" GELOESCH
T ! "
 635 PAUSE 1: PAUSE 0: GO TO 100
 700 REM UEBERSETZEN
 701 CLS : POKE adr,217: POKE ad
r+1,229: LET adr=adr+2
 702 FOR Z=1 TO P: INPUT INKEY$:
 PRINT (" " AND Z<100);(" " AND
Z<10);Z;" ";P$(Z)
 703 LET N$="": FOR X=1 TO 20
 704 IF P$(Z)(X)=CHR$ 32 THEN GO
```

```
 TO 710
 705 LET N$=N$+P$(Z)(X): NEXT X
 706 GO TO 9000
 710 FOR O=1 TO AN
 711 IF B$(O)( TO LEN N$)=N$ THE
N GO TO B(O)
 712 NEXT O
 713 LET ADR=RAM: PRINT #0;AT 1,
0;"BEFEHL NICHT VORHANDEN ": PAU
SE 1: PAUSE 0: GO TO 100
 720 NEXT Z: CLS
 721 POKE ADR,225: POKE ADR+1,21
7: LET ADR=ADR+2: PRINT #0;AT 1,
0;"KEINE FEHLER GEFUNDEN."
 722 PRINT AT 10,0;"SAVE ~NAME~
CODE ";RAM;",";ADR-RAM+1
 723 POKE adr,201
 724 PAUSE 1: PAUSE 0
 725 STOP
 800 REM SAVE
 801 CLS
 802 INPUT "NAME : ";N$: SAVE N$
+"0" DATA P$()
 803 DIM O(1): LET O(1)=P
 804 SAVE N$+"1" DATA O()
 805 DIM O(1): LET O(1)=ADR
 806 SAVE N$+"2" DATA O()
 807 VERIFY "" DATA P$()
 808 DIM O(1): LET O(1)=P
 809 VERIFY "" DATA O()
 810 DIM O(1): LET O(1)=ADR
 811 VERIFY "" DATA O()
 812 PRINT AT 10,0;"VERIFY OK !
": PAUSE 1: PAUSE 0: GO TO 100
 900 REM LOAD
 901 CLS
 902 INPUT "NAME : ";N$
 903 LOAD N$+"0" DATA P$()
 904 DIM O(1)
 905 LOAD N$+"1" DATA O()
 906 LET P=O(1)
 907 DIM O(1)
 908 LOAD N$+"2" DATA O()
 909 LET ADR=O(1)
 910 GO TO 100
1000 REM EINFUEGEN
1001 CLS
1002 INPUT "WELCHE ZEILE SOLL EI
NGEFUEGT    WERDEN ? ";ZEILE
1003 IF ZEILE>P THEN GO TO 1002
1004 IF ZEILE=INS THEN PRINT #0;
AT 1,0;"PROGRAMMSPEICHER VOLL !
": PAUSE 1: PAUSE 0: GO TO 100
1005 FOR O=P TO ZEILE STEP -1: L
ET P$(O+1)=P$(O): NEXT O: LET P=
P+1: LET P$(ZEILE)="
                    "
1006 PRINT #0;AT 1,0;"ZEILE EING
EFUEGT !": PAUSE 1: PAUSE 0: GO
TO 100
1500 REM POKE
1501 LET X=X+1: LET X$="": LET Y
$=""
1502 FOR O=X TO X+5: IF P$(Z)(O)
="," THEN GO TO 1510
1503 LET X$=X$+P$(Z)(O): LET L$=
P$(Z)(O): IF L$<"0" OR L$>"9" TH
EN GO TO 9000
1504 NEXT O: GO TO 9000
1510 LET X=O+1: FOR O=X TO X+3:
IF P$(Z)(O)=" " THEN GO TO 1520
1511 LET Y$=Y$+P$(Z)(O): LET L$=
P$(Z)(O): IF L$<"0" OR L$>"9" TH
EN GO TO 9000
1512 NEXT O: GO TO 9000.
1520 IF X$="" OR Y$="" THEN GO T
O 9000
1521 LET X1=VAL X$: LET Y1=VAL Y
$: IF X1>65535 OR Y1>255 THEN GO
 TO 9100
1522 RESTORE 1600
1523 FOR O=ADR TO ADR+4: READ Z1
: POKE O,Z1: NEXT O: LET ADR=ADR
+5: GO TO 720
1600 DATA 33,FN A(X1),FN B(X1),5
4,Y1
1750 REM PEEK
1751 LET X=X+1: LET X$="": FOR O
=X TO X+5: IF P$(Z)(O)=" " THEN
GO TO 1760
1752 LET X$=X$+P$(Z)(O): LET L$=
P$(Z)(O): IF L$<"0" OR L$>"9" TH
EN GO TO 9000
1753 NEXT O: GO TO 9000
1760 IF X$="" THEN GO TO 9000
1761 LET X1=VAL X$: IF X1>65535
THEN GO TO 9000
1762 RESTORE 1800: FOR O=ADR TO
ADR+8: READ Z1: POKE O,Z1: NEXT
O: LET ADR=ADR+9: GO TO 720
1800 DATA 33,FN A(X1),FN B(X1),7
8,6,0,205,27,26
2000 REM PLOT
```

```
2001 LET X=X+1
2002 LET X$="": LET Y$=""
2003 FOR O=X TO X+3
2004 IF P$(Z)(O)="," THEN GO TO
2010
2005 LET X$=X$+P$(Z)(O): LET L$=
P$(Z)(O): IF L$<>"," AND L$<>"-"
 AND L$<>"+" AND (L$<"0" OR L$>"
9") THEN GO TO 9000
2006 NEXT O
2007 GO TO 9000
2010 LET X=O+1
2011 FOR O=X TO X+3
2012 IF P$(Z)(O)=" " THEN GO TO
2020
2013 LET Y$=Y$+P$(Z)(O): LET L$=
P$(Z)(O): IF L$<>"," AND L$<>"-"
 AND L$<>"+" AND (L$<"0" OR L$>"
9") THEN GO TO 9000
2014 NEXT O
2015 GO TO 9000
2020 IF X$="" OR Y$="" THEN GO T
O 9000
2025 LET X1=VAL X$: LET Y1=VAL Y
$: IF X1<0 OR X1>255 OR Y1<0 OR
Y1>175 THEN GO TO 9100
2027 RESTORE 2050
2030 FOR O=ADR TO ADR+5
2040 READ Z1: POKE O,Z1
2041 NEXT O
2050 DATA 1,X1,Y1,205,229,34
2055 LET ADR=ADR+6
2056 GO TO 720
2250 REM DRAW
2251 LET X=X+1
2252 LET X$="": LET Y$=""
2253 FOR O=X TO X+4
2254 IF P$(Z)(O)="," THEN GO TO
2260
2255 LET X$=X$+P$(Z)(O): LET L$=
P$(Z)(O): IF L$<>"-" AND L$<>"+"
 AND L$<>"," AND (L$<"0" OR L$>"
9") THEN GO TO 9000
2256 NEXT O
2257 GO TO 9000
2260 LET X=O+1
2261 FOR O=X TO X+4
2262 IF P$(Z)(O)=" " THEN GO TO
2270
2263 LET Y$=Y$+P$(Z)(O): LET L$=
P$(Z)(O): IF L$<>"-" AND L$<>"+"
 AND L$<>"," AND (L$<"0" OR L$>"
9") THEN GO TO 9000
2264 NEXT O
2265 GO TO 9000
2270 IF X$="" OR Y$="" THEN GO T
O 9000
2271 LET X1=VAL X$: LET Y1=VAL Y
$: IF X1<-255 OR X1>255 OR Y1<-1
75 OR Y1>175 THEN GO TO 9100
2272 IF X1>=0 THEN LET Q1=1
2273 IF X1<0 THEN LET Q1=255
2274 IF Y1>=0 THEN LET W1=1
2275 IF Y1<0 THEN LET W1=255
2280 RESTORE 2400
2285 FOR O=ADR TO ADR+8
2286 READ Z1: POKE O,ABS Z1
2287 NEXT O
2288 LET ADR=ADR+9
2289 GO TO 720
2400 DATA 1,X1,Y1,17,Q1,W1,205,1
86,36
2500 REM CLS
2501 RESTORE 2600
2502 FOR O=ADR TO ADR+2
2503 READ Z1: POKE O,Z1
2504 NEXT O
2505 LET ADR=ADR+3
2506 GO TO 720
2600 DATA 205,107,13
2750 REM INIT
2751 LET X=X+1: LET X$=P$(Z)(X):
 IF X$<"0" OR X$>"3" THEN GO TO
9000
2752 LET X1=VAL X$
2775 RESTORE 2800
2776 FOR O=ADR TO ADR+4
2777 READ Z1: POKE O,Z1
2778 NEXT O
2779 LET ADR=ADR+5
2780 GO TO 720
2800 DATA 62,X1,205,1,22
3000 REM AT
3001 LET X=X+1
3002 LET X$="": LET Y$=""
3003 FOR O=X TO X+2
3004 IF P$(Z)(O)="," THEN GO TO
3010
3005 LET X$=X$+P$(Z)(O): LET L$=
P$(Z)(O): IF L$<>"," AND (L$<"0"
 OR L$>"9") THEN GO TO 9000
3006 NEXT O: GO TO 9000
3010 LET X=O+1: FOR O=X TO X+2:
IF P$(Z)(O)=" " THEN GO TO 3020
```

```
3011 LET Y$=Y$+P$(Z)(O): LET L$=
P$(Z)(O): IF L$<>"," AND (L$<"O"
 OR L$>"9") THEN GO TO 9000
3012 NEXT O: GO TO 9000
3020 IF X$="" OR X$="" THEN GO T
O 9000
3021 LET X1=VAL X$: LET Y1=VAL Y
$: IF X1>21 OR Y1>31 THEN GO TO
9000
3022 RESTORE 3100
3023 FOR O=ADR TO ADR+8
3024 READ Z1: POKE O,Z1: NEXT O
3025 LET ADR=ADR+9
3026 GO TO 720
3100 DATA 62,22,215,62,X1,215,62
,Y1,215
3250 REM PAUSE
3251 LET X=X+1: LET X$=""
3252 FOR O=X TO X+5: IF P$(Z)(O)
=" " THEN GO TO 3260
3253 LET X$=X$+P$(Z)(O): LET L$=
P$(Z)(O): IF L$<"O" OR L$>"9" TH
EN GO TO 9000
3254 NEXT O: GO TO 9000
3260 IF X$="" THEN GO TO 9000
3261 LET X1=VAL X$: IF X1>65535
THEN GO TO 9100
3262 RESTORE 3300
3263 FOR O=ADR TO ADR+5
3264 READ Z1: POKE O,Z1: NEXT O
3265 LET ADR=ADR+6: GO TO 720
3300 DATA 1,FN A(X1),FN B(X1),20
5,61,31
3500 REM PRINT$
3501 LET X=X+1
3502 FOR O=LEN TO 1 STEP -1: IF
P$(Z)(O)=CHR$ 34 THEN GO TO 3510
3503 NEXT O: PRINT #0;AT 1,0;"EN
DE-ZEICHEN IN PRINTZEILE ";Z;"
  NICHT GEFUNDEN ! ": PAUSE 1: P
AUSE O: LET ADR=RAM: GO TO 100
3510 LET O=O-1: LET OENDE=O: IF
O<X THEN GO TO 9000
3511 LET X$="": FOR O=X TO OENDE
: LET X$=X$+P$(Z)(O): NEXT O
3512 LET X1=LEN X$
3513 LET START=ADR+12: LET NEXT=
ADR+12+X1
3514 RESTORE 3600: FOR O=ADR TO
ADR+11: READ Z1: POKE O,Z1: NEXT
 O
3600 DATA 17,FN A(START),FN B(ST
ART),1,FN A(X1),FN B(X1),205,60,
32,195,FN A(NEXT),FN B(NEXT)
3601 LET I=1: FOR O=ADR+12 TO AD
R+11+X1
3602 POKE O,CODE (X$(I)): LET I=
I+1: NEXT O
3603 LET ADR=NEXT
3604 GO TO 720
3750 REM PRINT#
3751 LET X=X+1
3752 LET X$="": FOR O=X TO X+4:
IF P$(Z)(O)=" " THEN GO TO 3760
3753 LET X$=X$+P$(Z)(O): LET L$=
P$(Z)(O): IF L$<"O" OR L$>"9" TH
EN GO TO 9000
3754 NEXT O: GO TO 9100
3760 LET X1=VAL X$
3761 RESTORE 3800: FOR O=ADR TO
ADR+5: READ Z1: POKE O,Z1: NEXT
O: LET ADR=ADR+6: GO TO 720
3800 DATA 1,FN A(X1),FN B(X1),20
5,27,26
4000 REM PRINT%
4001 LET X=X+1
4002 LET X$=P$(Z)(X)
4003 LET X1=CODE X$
4004 RESTORE 4100: FOR O=ADR TO
ADR+2: READ Z1: POKE O,Z1: NEXT
O: LET ADR=ADR+3: GO TO 720
4100 DATA 62,X1,215
4250 REM INK
4251 RESTORE 4300: LET X=X+1: LE
T X$=P$(Z)(X): LET X1=VAL X$: FO
R O=ADR TO ADR+11: READ Z1: POKE
 O,Z1: NEXT O: LET ADR=ADR+12: G
O TO 720
4300 DATA 205,77,13,62,16,215,62
,X1,215,205,173,28
4500 REM PAPER
4501 RESTORE 4600: LET X=X+1: LE
T X$=P$(Z)(X): LET X1=VAL X$: FO
R O=ADR TO ADR+11: READ Z1: POKE
 O,Z1: NEXT O: LET ADR=ADR+12: G
O TO 720
4600 DATA 205,77,13,62,17,215,62
,X1,215,205,173,28
4750 REM BORDER
4751 RESTORE 4800: LET X=X+1: LE
T X$=P$(Z)(X): LET X1=VAL X$: FO
R O=ADR TO ADR+7: READ Z1: POKE
O,Z1: NEXT O: LET ADR=ADR+8: GO
TO 720
```

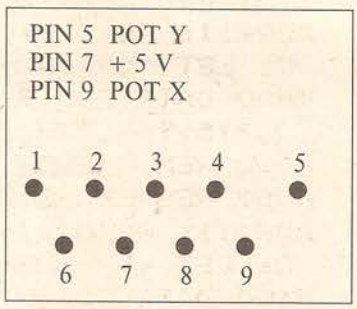
```
4800 DATA 205,77,13,62,X1,205,15
5,34
5000 REM COPY
5001 RESTORE 5100: FOR O=ADR TO
ADR+5: READ Z1: POKE O,Z1: NEXT
O: LET ADR=ADR+6: GO TO 720
5100 DATA 243,6,192,205,175,14
5250 REM QUER1
5251 RESTORE 5300: FOR O=ADR TO
ADR+3: READ Z1: POKE O,Z1: NEXT
O: LET ADR=ADR+4: GO TO 720
5300 DATA 253,203,87,206
5500 REM QUER0
5501 RESTORE 5600: FOR O=ADR TO
ADR+3: READ Z1: POKE O,Z1: NEXT
O: LET ADR=ADR+4: GO TO 720
5600 DATA 253,203,87,142
5750 REM INVERSE1
5751 RESTORE 5800: FOR O=ADR TO
ADR+3: READ Z1: POKE O,Z1: NEXT
O: LET ADR=ADR+4: GO TO 720
5800 DATA 253,203,87,222
6000 REM INVERSE0
6001 RESTORE 6100: FOR O=ADR TO
ADR+3: READ Z1: POKE O,Z1: NEXT
O: LET ADR=ADR+4: GO TO 720
6100 DATA 253,203,87,158
6250 REM HOME
6251 RESTORE 6300: FOR O=ADR TO
ADR+5: READ Z1: POKE O,Z1: NEXT
O: LET ADR=ADR+6: GO TO 720
6300 DATA 1,33,24,205,217,13
6500 REM SCROLL
6501 RESTORE 6600: FOR O=ADR TO
ADR+2: READ Z1: POKE O,Z1: NEXT
O: LET ADR=ADR+3: GO TO 720
6600 DATA 205,254,13
6750 REM CIRCLE
6751 LET X=X+1: LET X$="": LET Y
$="": LET Z$=""
6752 FOR O=X TO X+3: IF P$(Z)(O)
="," THEN GO TO 6760
6753 LET X$=X$+P$(Z)(O): LET L$=
P$(Z)(O): IF L$<"0" OR L$>"9" TH
EN GO TO 9000
6754 NEXT O: GO TO 9000
6760 LET X=O+1: FOR O=X TO X+3:
IF P$(Z)(O)="," THEN GO TO 6770
6761 LET Y$=Y$+P$(Z)(O): LET L$=
P$(Z)(O): IF L$<"0" OR L$>"9" TH
EN GO TO 9000
6762 NEXT O: GO TO 9000
6770 LET X=O+1: FOR O=X TO X+2:
IF P$(Z)(O)=" " THEN GO TO 6780
6771 LET Z$=Z$+P$(Z)(O): LET L$=
P$(Z)(O): IF L$<"0" OR L$>"9" TH
EN GO TO 9000
6772 NEXT O: GO TO 9000
6780 LET XX1=VAL X$: LET YY1=VAL
 Y$: LET ZZ1=VAL Z$: IF XX1>255
OR YY1>175 OR ZZ1>88 THEN GO TO
9100
6781 RESTORE 6800: FOR O=ADR TO
ADR+17: READ Z1: POKE O,Z1: NEXT
 O: LET ADR=ADR+18: GO TO 720
6800 DATA 62,XX1,205,40,45,62,YY
1,205,40,45,62,ZZ1,205,40,45,205
,45,35
7000 REM FLASH1
7001 RESTORE 7100: FOR O=ADR TO
ADR+11: READ Z1: POKE O,Z1: NEXT
 O: LET ADR=ADR+12: GO TO 720
7100 DATA 205,77,13,62,18,215,62
,1,215,205,173,28
7250 REM FLASH0
7251 RESTORE 7300: FOR O=ADR TO
ADR+11: READ Z1: POKE O,Z1: NEXT
 O: LET ADR=ADR+12: GO TO 720
7300 DATA 205,77,13,62,18,215,62
,0,215,205,173,28
7500 REM BRIGHT1
7501 RESTORE 7600: FOR O=ADR TO
ADR+11: READ Z1: POKE O,Z1: NEXT
 O: LET ADR=ADR+12: GO TO 720
7600 DATA 205,77,13,62,19,215,62
,1,215,205,173,28
7750 REM BRIGHT0
7751 RESTORE 7800: FOR O=ADR TO
ADR+11: READ Z1: POKE O,Z1: NEXT
 O: LET ADR=ADR+12: GO TO 720
7800 DATA 205,77,13,62,19,215,62
,0,215,205,173,28
8000 REM FLASH8
8001 RESTORE 8100: FOR O=ADR TO
ADR+11: READ Z1: POKE O,Z1: NEXT
 O: LET ADR=ADR+12: GO TO 720
8100 DATA 205,77,13,62,18,215,62
,8,215,205,173,28
8250 REM BRIGHT8
8251 RESTORE 8300: FOR O=ADR TO
ADR+11: READ Z1: POKE O,Z1: NEXT
 O: LET ADR=ADR+12: GO TO 720
8300 DATA 205,77,13,62,19,215,62
,8,215,205,173,28
```

```
8500 REM BREAK
8501 RESTORE 8600: LET X1=ADR: L
ET Y1=ADR+9: FOR O=ADR TO ADR+8:
 READ Z1: POKE O,Z1: NEXT O: LET
 ADR=ADR+9: GO TO 720
8600 DATA 205,84,31,210,FN A(Y1)
,FN B(Y1),195,FN A(X1),FN B(X1)
8750 REM KEY
8751 LET X=X+1: LET X$=P$(Z)(X)
8752 IF (X$<" " OR X$>CHR$ 90) A
ND (X$<"^" OR X$>CHR$ 122) THEN
PRINT #0;AT 1,0;"CODE AUSSERHALB
 DES ERLAUBTEN   BEREICHES IN ";
Z: LET ADR=RAM: PAUSE 1: PAUSE 0
: GO TO 100
8753 RESTORE 8800: LET X1=CODE X
$: LET Y1=ADR+1: FOR O=ADR TO AD
R+10: READ Z1: POKE O,Z1: NEXT O
: LET ADR=ADR+11: GO TO 720
8800 DATA 175,255,58,8,92,254,X1
,194,FN A(Y1),FN B(Y1),175
8900 REM DI
8901 POKE ADR,243: LET ADR=ADR+1
: GO TO 720
8910 REM EI
8911 POKE ADR,251: LET ADR=ADR+1
: GO TO 720
8999 REM FEHLERMELDUNGEN
9000 LET ADR=RAM: INPUT INKEY$:
INVERSE 1: PRINT (" " AND Z<100)
;(" " AND Z<10);Z;" ";P$(Z): INV
ERSE 0: PRINT #0;AT 1,0;"FALSCHE
 EINGABE IN ";Z: PAUSE 1: PAUSE
0: GO TO 100
9100 LET ADR=RAM: INPUT INKEY$:
INVERSE 1: PRINT (" " AND Z<100)
;(" " AND Z<10);Z;" ";P$(Z): INV
ERSE 0: PRINT #0;AT 1,0;"ZAHL AU
SSERHALB DES BEREICHES IN";Z: PA
USE 1: PAUSE 0: GO TO 100
9101 REM FEHLERMELDUNGEN/ENDE
9998 REM PROGRAMM SPEICHERN
9999 PAUSE 1: PAUSE 50: POKE 237
36,181: SAVE "GCOMPILER" LINE 1:
 GO TO 9999
```

Der C-64 als Speicher-Oszilloskop

Das folgende Maschinenprogramm ermöglicht die Abspeicherung von rund 8000 Meßwerten, die mit einem der A/D-Wandler bestimmt wurden. Die gemessenen Widerstandsschwankungen werden – direkt während der Messung – graphisch als Funktion der Zeit dargestellt.

Die graphische Darstellung einer Meßreihe gibt die Ergebnisse viel anschaulicher wieder als eine abstrakte Zahlenkolonne. Da der C-64 jedoch keine X-Y-Punktgrafik besitzt, ist einiges an Software notwendig, um die gemessene Funktion auf dem Bildschirm darzustellen. Als Eingang für anzulegende Meßeinrichtungen werden die zwei A/D-Wandler an Controlport 1 verwendet. Mit Hilfe dieser Wandler wird ein analoges Signal in einen digitalen Zahlenwert umgewandelt. Bei den 2 A/D-Wandlern des C-64 werden Potentiometeränderungen gemessen. Die folgende Abbildung zeigt die Lage der 2A/D-Eingänge.



Möchte man den Oszillographensimulator für eigene Anwendungen nutzen, dann muß man zwischen Pin 7 und Pin 9 eine von der Meßeinrichtung produzierte Widerstandsschaltung anlegen. Von Pin 7 werden hierfür +5 Volt bereit gestellt. Der gemessene digitale Wert des Widerstandes befindet sich in Register 25 (Cont.1) und Register 26 (Cont.2) des SID. Adresse 54297 und 54298. Um die ganze Breite des Registers auszunutzen, muß sich der gemessene Widerstand im Bereich von 0-500 Kiloohm bewegen. Bei 256 möglichen Zuständen des Registers beträgt 1 digitaler Schritt also +- 500/256 = 2 Kiloohm. Der Inhalt dieses Registers wird vom Programm zur Auslenkung in Y-Richtung benutzt. Da die Auflösung in Y-Richtung im Hires Mode 200 Punkte beträgt, sind für das Programm Widerstandsschwankungen von 0 bis 400 Kiloohm relevant.

Nun zum Programm selbst. Jeder der schon einmal mit High-Res. Graphik experimentiert hat, weiß, wie umständlich es ist, in Maschinensprache aus den Koordinaten des Punktes seine Adresse auf dem Bildschirm und seine Position innerhalb des Bytes zu berechnen. Um möglichst schnell zu arbeiten, verwendet dieses Programm nicht den herkömmlichen Algorythmus zur Bestimmung der Punktposition. Statt diesem wird eine Tabelle benutzt, die mit geringem Aufwand die Bestimmung der richtigen Adresse und die Position des Punktes innerhalb des Bytes, gestattet. In Basic hingegen wäre die ganze Angelegenheit sehr viel einfacher aber dafür auch tödlich langsam. Nur der

A/D-Wandler bringt mit seinem Meßwert alle 512 Taktzyklen auf den neuesten Wert. Ein Maschinenprogramm kann also in der Zwischenzeit in aller Ruhe die Position des Punktes auf dem Bildschirm berechnen, und ihn setzen. Dieser Vorgang würde mit dem vorliegenden Programm ca. 100 Taktzyklen benötigen. In den restlichen 400 Taktzyklen werden daher noch andere Unterprogramme ausgeführt, welche die Anwendung des Oszilloskops komfortabel gestalten. Um die Funktionsweise des Programms erproben zu könne, schließt man am besten bei Controlport 1 zwischen Pin 7 und Pin 9 einen Fotowiderstand oder ein Regelpotentiometer an, (Widerstandsbereich zwischen 0 und 400 kOhm).

Der Anschluß des Widerstandes erfolgt am besten mit Hilfe eines an den Controlport passenden Steckers. Nachdem das Maschinenprogramm in seinen Speicherbereich geschrieben wurde, startet man das Eingabemenü. Auf dem Bildschirm erscheint die Frage nach der Zeitbasis. Hier kann man angeben, wie lange ein Bildschirmdurchlauf dauern soll. Es können Werte von 0 bis 17 sec (z. B. 10.5) eingegeben werden. Bei der Eingabe von 0 arbeitet das Programm mit der Maximalgeschwindigkeit des A/D-Wandlers. Alle 0.5 msec wird ein Punkt gesetzt. Ein Bildschirmdurchlauf dauert dann 0.5 * 256 punkte = 128 msec., also ca. 1/8 sec. Nach Eingabe der Zeitbasis wird eine vom Widerstand abhängige Kurve gezogen. Falls der angeschlossene Widerstand > 400 kOhm ist, wird am oberen Bildrand entlang nur eine Linie gezogen. Drücken von F1 bewirkt das Überblenden eines Gitters, erneutes Drücken löscht es wieder. Hat der Oszillographenstrahl das Bildschirmende erreicht, wird das Gitter ebenfalls gelöscht. F3 bewirkt ein Anhalten des Strahles. F4 läßt ihn wieder weiterlaufen. Bei Verwendung von F5 bemerkt man zunächst ein Umsetzen der Rahmenfarbe. Dies deutet an, daß alle nun folgenden Werte in einem Puffer abgespeichert werden. Mit F6 kann der Anspeichervorgang unterbrochen werden. Eine erneute Verwendung von F5 fügt die nachfolgenden Meßwerte direkt an die bereits gespeicherten Werte an. Der Puffer umfaßt 32 Bildschirmseiten. Wenn der Puffer aufgefüllt ist, wird automatisch unterbrochen. Mit F7 läßt sich der zweite Oszillographenstrahl einschalten. Er verarbeitet die Widerstandswerte an Controlport 1 POTY. Durch F8 wird Strahl 2 wieder ausgeschaltet. Achtung, wird im 2-Strahlmodus mit F5 abgespeichert, so gelangen nur die Meßwerte des A/D-Wandlers von Controlport 2 in den Puffer. Durch die SPACE-Taste kann der Programmablauf beendet werden. Auf dem Bildschirm erscheint dann Restart or Data processing. Drückt man R startet das Programm neu. Mit D können die Werte im Puffer aufgearbeitet werden. Es erscheinen dann die Fragen:

Print Page?

Clear map?

Bei Print Page wird die Speicherseite im Puffer angegeben, die dargestellt werden soll. Clear map fragt, ob der vorherige Inhalt der Bit map gelöscht werden soll.

Bei no wird die Speicherpage mit dem Inhalt der Bit map überlagert. Auch hier kann mit F1 ein Gitter übergeblendet oder wieder gelöscht werden. Aus praktischen Gründen werden Bereiche in denen sich Kurve und Gitter überlagern, revers dargestellt. Mit den Tasten CRSR right und CRSR left kann man von der dargestellten Page aus im Speicher nach rechts oder links scrollen. Mit Hilfe der SPACE-Taste kann die Fläche unterhalb des auf dem Bildschirm dargestellten Kurvenausschnittes angezeigt werden. Durch nochmaliges Drücken der SPACE-Taste gelangt man wieder zu Restart or Data processing.

Hier noch einmal zusammengefaßt die Bedeutung der Tasten im Direkt- oder Data Processing Modus.

Direkt

F1	Gitter an/aus
F3	Strahl stop
F4	Strahl weiter
F5	Speicher an
F6	Speicher aus
F7	2-Strahl an
F8	2-Strahl aus
SPACE	unterbrechen

Data processing Modus

F1	Gitter an/aus
CRSR right	Bild nach rechts verschieben
CRSR left	Bild nach links verschieben
SPACE	Fläche unterhalb der dargestellten Kurve

Teil 1 - LOADER

```
C64 LISTING MIT CHECKSUMMEN (C V1.0)

10 FORI= 49152 TO 49920 :READ DA:POKEI,D
A:NEXTI                                  <238>
100 DATA 76,167,192,76,200,192,76,226,19
2,76,27,192,76,193,193,76,142            <44>
101 DATA 192,76,4,194,76,219,193,76,168,
194,234,169,32,133,176,169,96            <10>
102 DATA 133,177,169,0,133,178,169,202,1
33,179,169,0,133,180,169,203,133         <17>
103 DATA 181,169,57,133,186,169,1,133,18
7,160,26,132,191,160,0,169,7             <132>
104 DATA 24,101,176,133,176,169,30,101,1
77,133,177,32,121,192,198,191,240        <37>
105 DATA 45,162,7,56,165,176,233,1,133,1
76,176,2,198,177,32,121,192              <58>
106 DATA 202,208,239,165,176,56,229,186,
133,176,165,177,229,187,133,177,76       <216>
107 DATA 79,192,165,176,145,178,165,177,
145,180,200,96,56,169,0,162,8            <213>
108 DATA 42,149,167,202,208,250,169,0,13
3,176,169,64,133,177,160,0,162           <14>
109 DATA 4,169,192,145,176,200,208,251,2
30,177,202,208,246,96,169,96,133         <69>
110 DATA 87,169,59,141,17,208,173,0,221,
41,252,9,2,141,0,221,169                 <46>
111 DATA 8,141,24,208,169,0,133,251,169,
68,133,252,96,234,169,0,133              <217>
112 DATA 182,169,96,133,183,162,32,160,0
,169,0,145,182,200,208,251,230           <187>
113 DATA 183,202,208,246,96,162,0,134,18
4,172,25,212,192,199,144,2,160           <15>
114 DATA 199,132,185,32,147,193,173,222,
3,240,14,172,26,212,192,199,144          <244>
115 DATA 2,160,199,132,185,32,147,193,17
3,227,3,240,3,32,60,194,32               <18>
116 DATA 66,241,240,74,201,32,240,93,201
,133,208,3,32,219,193,201,134            <198>
117 DATA 208,3,32,86,194,201,135,208,15,
160,0,140,224,3,160,4,140                <228>
118 DATA 32,208,72,32,123,193,104,201,13
9,208,15,160,1,140,224,3,160             <107>
119 DATA 0,140,32,208,72,32,135,193,104,
201,136,208,5,160,1,140,222              <230>
120 DATA 3,201,140,208,5,160,0,140,222,3
,230,184,208,132,173,224,3               <126>
121 DATA 208,8,230,252,165,252,197,87,17
6,6,32,200,192,76,226,192,32             <10>
122 DATA 24,229,76,163,253,165,251,56,22
9,184,133,251,176,2,198,252,96           <121>
123 DATA 165,251,24,101,184,133,251,144,
246,230,252,96,166,184,164,185,185       <33>
124 DATA 0,203,133,187,138,41,248,24,121
,0,202,133,186,144,2,230,187             <112>
```

```
125 DATA 138,41,7,170,160,0,181,168,17,1
86,145,186,174,224,3,208,6                <102>
126 DATA 164,184,165,185,145,251,96,234,
169,0,133,176,169,68,133,177,162          <49>
127 DATA 28,160,0,169,0,145,176,200,208,
251,230,177,202,208,246,96,234            <193>
128 DATA 169,0,133,176,169,96,133,177,16
0,0,162,7,169,1,81,176,145                <8>
129 DATA 176,200,202,208,246,169,255,81,
176,145,176,200,208,235,230,177,165       <26>
130 DATA 177,201,128,208,227,96,234,173,
229,3,24,105,68,133,179,173,228           <126>
131 DATA 3,133,178,160,0,177,178,133,185
,132,184,32,147,193,164,184,200           <88>
132 DATA 208,242,32,66,241,201,133,208,3
,32,219,193,201,32,208,9,32               <69>
133 DATA 168,194,32,24,229,76,163,253,76
,94,194,234,169,0,141,6,220               <62>
134 DATA 173,227,3,141,7,220,169,9,141,1
5,220,173,6,220,205,6,220                 <82>
135 DATA 208,248,96,32,66,241,201,138,20
8,249,96,234,201,29,240,7,201             <72>
136 DATA 157,240,32,76,34,194,24,169,32,
109,228,3,141,228,3,144,45                <39>
137 DATA 238,229,3,173,229,3,201,31,144,
35,206,229,3,141,229,3,240                <191>
138 DATA 27,56,173,228,3,233,32,141,228,
3,176,16,206,229,3,173,229                <245>
139 DATA 3,16,8,238,229,3,169,0,141,228,
3,32,200,192,76,4,194                     <24>
140 DATA 234,169,0,133,180,133,181,133,1
82,160,0,177,178,200,240,11,24            <92>
141 DATA 101,180,133,180,144,244,230,181
,208,240,96,4,255,0,251,46,238            <227>
142 DATA 0,255,4,255,0,255,3,255,0,255,0
,255,0,255,34,255,0                       <113>
143 DATA 255,20,8,0,111,0,255,0,255,4,25
1,6,186,0,255,0,254                       <136>
144 DATA 4,42,11,40,2,191,10,43,6,191,55
,190,2,191,10,190,38                      <95>
145 DATA 46,111,64,254                    <83>
  ENDE DES LISTINGS
   [DRUCK:F.B.]
```

Teil 2 - Hauptprogramm

```
5010 POKE56,4*16:CLR                      <45>
5020 A= 49152:POKE992,1:POKE53280,0       <130>
5025 POKE990,0                            <36>
5027 POKE991,1                            <48>
5040 POKE53281,0:POKE53280,0              <12>
5050 PRINT"{CLEAR DOWN10}"                <228>
5060 INPUT"{SPACE7}TIMEBASE{SPACE}IN{SPA
CE RVSON}SEC{RVSOFF}";T                   <97>
5065 T=T*255/17                           <50>
5067 IF T>255 THEN5025                    <87>
5070 POKE995,T                            <61>
6000 POKE53248+17,0                       <34>
6005 SYSA+9:SYS A+12                      <40>
6010 SYSA+3:SYS A:SYS A+6                 <214>
6020 POKE53281,0:POKE53280,0              <228>
6030 PRINT"{CLEAR DOWN10 PURPLE RIGHT6 R
VSON}R{RVSOFF}ESTART{SPACE}OR{SPACE RVSO
N}D{RVSOFF}ATA{SPACE}PROCESSING"          <167>
6040 WAIT 198,1:GET B$                    <136>
6050 IF B$="R"THEN 5020                   <219>
6060 IF B$<>"D"THEN 6040                  <34>
6070 PRINT"{CLEAR DOWN10 PURPLE}"         <142>
6090 PRINT"{SPACE2}PRINT{SPACE}PAGE{SPAC
E4}"                                      <96>
7000 PRINT                                <12>
7010 PRINT                                <22>
7020 PRINT                                <32>
7030 PRINT"{SPACE2}CLEAR{SPACE}MAP{SPACE
5}"                                       <99>
7040 PRINT"{HOME DOWN10 CYAN}"            <6>
7050 INPUT"{SPACE2}PRINT{SPACE}PAGE{SPAC
E4}";Z                                    <14>
7060 PRINT                                <72>
7070 PRINT                                <82>
7080 PRINT                                <92>
7090 INPUT"{SPACE2}CLEAR{SPACE}MAP{SPACE
}Y/N{SPACE}";C$                           <19>
8000 IF Z>32THEN 6070                     <42>
8020 IF C$="Y"THEN SYS A+3                <104>
8030 POKE992,1:POKE997,Z:POKE996,0        <177>
8040 POKE 53248+17,0                      <34>
8050 SYS A+15:SYS A:SYS A+18              <184>
8060 POKE53281,0:POKE53280,0              <228>
8090 H=PEEK(181):L=PEEK(180)              <166>
9000 E= H*256+L                           <42>
9010 PRINT"{DOWN9 SPACE9}FLAECHE{SPACE}=
{SPACE2}";E                               <38>
9020 WAIT198,1:GET A$:IF A$="{SPACE}"THE
N 6030                                    <199>
  ENDE DES LISTINGS
   [DRUCK:F.B.]
```

Der Apple wird zur Supersoundmaschine

RGS-Digital-Synthesizer für Ihren Apple-Computer

Achtung: Dieses Programm ist nur lauffähig mit dem Digital Sound Editor aus Ausgabe 9

Mit diesem Programm wird es möglich, die mit DSE (CPU Ausgabe Sept. 85) erzeugten Sounds wie auf einem Synthesizer zu spielen. Damit wird der Apple zu einer Supersoundmaschine. Das Programm eignet sich besonders für Spezialtoneffekte. Man kann z.B. ein kurzes Wort mit DSE digitalisieren und dann mit dem Synthesizer durch alle Tonhöhen 'jagen'. Durch die 'REPT'-Taste wird es möglich, den Sound zu 'scratchen'. Am besten testet man das alles mal aus, der Synthesizer bietet einfach unendlich viele Soundvariationen, da man Sounds auch wie beim DSE-Programm mixen kann. Aber nicht nur 2, sondern ganze 10 Sounds! Diese 10 Sounds können sich auf einmal im Speicher befinden und man kann dann mit den Tasten 1,2,3...0 zwischen ihnen umschalten.

Der Synthesizer muß immer mit dem Startprogramm 'SYNTHESIZER.START' gestartet werden (wegen der Verlagerung des Basicprogrammspeichers). Nach dem Start erscheint auf dem Bildschirm ein Menü. Zuerst sollte man einige Sounds mit der Funktion <2> „Sound laden" in den Speicher bringen. Nach Anwählen dieser Funktion wird man nach einem (Speicher)Bereich (1-10) gefragt. Man kann dann den ersten Sound in Bereich 1 laden, den zweiten in Bereich 2 usw.. Im PLAY-MODUS <1> kann mit den Zahlentasten 1-0 zwischen den einzelnen Bereichen umgeschaltet werden. Die Pfeiltasten erhöhen oder erniedrigen die Tonlage des Synthesizers. 'M' setzt die Mixfunktion in Gang. Man wird dann gefragt, welche Bereiche man mixen will, dabei sind 2 Zahlen anzugeben, die erste Zahl für den Anfang und die zweite für das Ende des Speicherbereichs. Auch hier ist der Zahlenbereich 1-10.

'X' schatet den Mixmodus wieder ab und 'ESC' ruft das Menü wieder auf.

Hinweise zur Programmeingabe:

1. DSE.OBJ1 laden (CPU-Ausgabe September'85) und ab $4100 veränderten Hexdump eingeben.
2. Absaven mit 'BSAVE DIGSYN.OBJ1,A$4000,L$157
3. Basic Startprogramm eingeben und mit 'SAVE SYNTHESIZER.START' saven.
4. Basic Hauptprogramm eingeben und mit 'SAVE DIGITAL SYNTHESIZER' saven.
5. Startprogramm starten.

Vorprogramm:

```
10  HOME : PRINT "DIGITAL SYNTHESIZER BY RGS IS LOADING...";
20  POKE 104,96: POKE 256 * 96,0
25  PRINT "----------------------------------------";: PRINT
30  PRINT  CHR$ (4);"BLOAD DIGSYN.OBJ1"
40  PRINT  CHR$ (4);"RUN DIGITAL SYNTHESIZER"
```

BASIC-HAUPTPROGRAMM:

```
10  REM ************************
15  REM *                      *
20  REM * DIGITAL SYNTHESIZER *
25  REM * ^^^^^^^^^^^^^^^^^^^^ *
30  REM *                      *
35  REM * WRITTEN 1985 BY:     *
40  REM *                      *
45  REM * ROLAND GRAF          *
50  REM * KESSELEWEG 20        *
55  REM * 7473 STRASSBERG 1    *
60  REM *                      *
65  REM ************************
70  REM
100  HOME
110 :
120  REM ** VARIABLEN **
130 :
140 SP = 16640:L$ = "                "
150 BR = 1:M = 0
160 GF = 3:A = 253:E = 254:G = 255
170  DIM T(90)
180 T(65) = 17:T(87) = 16:T(83) = 15:T(69) = 14
190 T(68) = 13:T(70) = 12:T(84) = 11:T(71) = 10
200 T(89) = 9:T(72) = 8:T(85) = 7:T(74) = 6
210 T(75) = 5:T(79) = 4:T(76) = 3:T(80) = 2
220 T(59) = 1
230  FOR I = 1 TO 9
240 T(48 + I) = I + 19
250  NEXT
260 T(48) = 29:T(77) = 33
270 T(27) = 30:T(88) = 34
280 T(8) = 31:T(21) = 32: REM  <--/-->
290 :
300  REM ** MENUE **
310 :
320  INVERSE : PRINT "++++++++++++++++++++++++++++++++++++++++++++";: PRINT "+++++++
 RGS-DIGITAL-SYNTHESIZER ++++++++";: PRINT "++++++++++++++++++++++++++++++++++++
++++";: NORMAL
340  PRINT : PRINT "OPTIONEN:"
350  PRINT : PRINT "(1) PLAY-MODUS"
360  PRINT "(2) SOUND LADEN"
370  PRINT "(3) ENDE"
380  VTAB 11: HTAB 1: PRINT "( )": VTAB 11: HTAB 2: GET IN$
390  IF IN$ < "1" OR IN$ > "3" THEN 380
400  IF IN$ = "1" THEN 1000
410  IF IN$ = "2" THEN 2000
420  HOME : END
1000 :
1010  REM ** PLAY MODUS **
1020 :
1030  HOME : POKE A,80: POKE E,81: POKE G,17
1035 BR = 1
1040  INVERSE
1050  PRINT "  RGS DIGITAL SYNTHESIZER - PLAY MODUS  ";
1060  NORMAL
1070  PRINT : PRINT "MANUAL:"
1080  PRINT : PRINT "..W.E...T.Y.U...O.P...": PRINT ".A.S.D.F.G.H.J.K.L.;.."
1090  PRINT : PRINT "..C#D#..F#G#A#..C#D#..": PRINT ".C.D.E.F.G.A.H.C.D.E.."
1100  VTAB 12: HTAB 1: PRINT "FAKTOR:";GF: VTAB 14: HTAB 1: PRINT "BEREICH:";BR
```

HALLO

„Gewerbliche“: Dieser „RAUM“ ist zu verkaufen! Bitte Media-Unterlagen anfordern!

```
1110  VTAB 16: HTAB 1: INVERSE : PRINT "  RGS DIGITAL SYNTHESIZER - PLAY MODUS
";: NORMAL
1120  VTAB 18: HTAB 1: PRINT "ESC = EXIT": PRINT "<- TON HOEHER": PRINT "-> TON
TIEFER": PRINT "M = MIXEN "
1121  PRINT "X = MIXEN AUS"
1123  IF MI = 1 THEN  VTAB 13: HTAB 15: PRINT "MIX ON "
1124  IF MI = 0 THEN  VTAB 13: HTAB 15: PRINT "MIX OFF"
1125  GOTO 1140
1130  CALL SP
1140  GET A$
1150 X = T( ASC (A$)): IF X > 19 THEN 1500
1155  IF X = 0 THEN 1140
1160  POKE A,79 + BR + M1: POKE E,80 + BR + M2: POKE G,X + GF
1165  IF MI = 1 THEN  POKE A,80 + M1: POKE E,81 + M2
1170  GOTO 1130
1500  REM ** SPECIAL KEYS **
1510  IF X = 30 THEN  HOME : GOTO 300
1515  IF X = 34 THEN M1 = 0:M2 = 0:MI = 0: GOTO 1000
1520  IF X = 31 AND GF > 0 THEN GF = GF - 1
1530  IF X = 32 AND GF < 100 THEN GF = GF + 1
1560  IF X > 19 AND X < 30 THEN 1570
1565  GOTO 1600
1570 BR = X - 19
1600  VTAB 12: HTAB 1: PRINT L$: VTAB 14: HTAB 1: PRINT L$: VTAB 12: HTAB 1: PRI
NT "FAKTOR:";GF: VTAB 14: HTAB 1: PRINT "BEREICH:";BR
1610  IF X = 33 THEN 1700
1620 X = 17
1630  GOTO 1160
1700  REM ** MIX **
1705 MI = 1
1710  HOME : PRINT "++ MIXEN ++"
1720  PRINT : INPUT "VON,BIS:";M1$,M2$
1730 M1 =  VAL (M1$):M2 =  VAL (M2$)
1740  IF M1 < 1 OR M1 > 10 THEN 1710
1745  IF M2 < 1 OR M2 > 10 THEN 1710
1750  IF M1 > M2 THEN 1710
1760 M1 = M1 - 1:M2 = M2 - 1
1770  GOTO 1000
2000  REM ** LADEN **
2010  HOME : PRINT "++ LADEN ++"
2020  PRINT : INPUT "IN BEREICH (1-10):";BE$
2025 BE =  VAL (BE$)
2030  IF BE < 1 OR BE > 10 THEN 2010
2040  PRINT : INPUT "FILENAME (RET=CATALOG):";FI$
2050  IF FI$ = "" THEN  PRINT  CHR$ (4);"CATALOG": GOTO 2040
2060 B = BE * 100 - 100:B = B + 5000
2070  PRINT  CHR$ (4);"BLOAD";FI$;",A$";B
2080  HOME : GOTO 300
```

Hex Dump Modifikation

```
]CALL-151
*4100.4157

4100- A9 01 85 F9 A2 00 A9 00
4108- 85 FB A5 FD 85 FC A1 FB
4110- 25 F9 C5 FA 85 FA F0 06
4118- 2C 30 C0 4C 23 41 EA EA
4120- 4C 23 41 20 7E 40 18 A5
4128- FB 69 01 85 FB A5 FC 69
4130- 00 85 FC C5 FE D0 D7 AD
4138- 00 C0 C9 00 30 13 A5 F9
4140- C9 80 F0 06 0A 85 F9 4C
4148- 0A 41 4C 00 41 FF FF FF
4150- 00 60 00 00 00 00 00 00
*
```

❃ Tips + Tricks für den C-64 ❃ Tips + Tricks für den C-64 ❃

Diese kleine Maschinenroutine erweitert das Basic um den Befehl GOTO X. Dieser dient dazu, bei Sprüngen mit einer berechneten Sprungadresse zu arbeiten statt mit einer festen.

Das Programm ist als Basic-Lader gestaltet und kann auch in eigene Programme eingebunden werden. Da die Routine frei verschiebbar ist, muß man nach dem Start des Laders zunächst eine Startadresse eingeben, ab der das Maschinenprogramm im Speicher abgelegt werden soll. Man kann hier z.B. den freien Bereich ab Adresse 704 verwenden, der für Sprite 11 reserviert ist. Der Aufruf erfolgt mit SYSa,x ; a steht hierbei für die gewählte Startadresse, x für einen beliebigen mathematischen Ausdruck, der die gewünschte Sprungadresse ergibt.

Listing siehe Seite 79

Ralph Lässig

ATARI

Disketten - schnell und komfortabel kopiert

ATARI

Super-Sektor-Kopierer

Der Super Sektor Kopierer ist ein Programm für jeden Atari Computer mit mindestens 48k RAM und einer Diskettenstation.

Es unterstützt Single Density der 810,105o und anderer Stationen. Außerdem kann mit ihm DOS unabhängig, ein Duplikat einer Diskette im 1050 Format angefertigt werden. Dieses Format wird von DOS 3, Sparta DOS und DOS 2.5 sowie einiger modifizierten DOS 2 Versionen benutzt. So kann man ohne die Nachteile einer DOS-Kopie eine Diskette kopieren deren RAM zerstört worden ist. (BAM=VTOC=Block Allocation Map=Sector Belegungs Tabelle).

Nach dem Eingeben des Listings (ob Source oder Data) wird ein Programm auf Diskette geschrieben mit einer Start-Adresse für OS/A+ DOS Versionen. Falls man das Programm auf anderen DOS-Versionen laufen lassen möchte, ist folgendes zu tun. Laden (mit Binär LOAD im DOS) und dann: DOS 2: K (return) Scopy.-Com,aaoo,b700,aaoo,aaoo nach dem Laden eingeben, und bei DOS 3: S Scopy.com(return)aaoo(return)b7oo(return)aaoo(return)aaoo(return) eingeben. Nun kann das Programm über das DOS geladen werden. Es startet automatisch.

Das Programm ist vollkommen menügesteuert. Die Eingaben erfolgen durch den Joystick. Auf dem Bildschirm ist eine „Maus" sichtbar. Am Anfang wird ein Status Request ausgeführt, der feststellt, welche Laufwerke angeschlossen sind. Das kann man mit Reset oder dem Menüpunkt „Status Request" wiederholen. Danach wird ein optisches Menü ausgegeben. Es enthält einige Felder, die den Bereich kennzeichnen, wo man mit der Maus, welche Reaktion erhält. In der ersten Reihe stehen die Laufwerke, die man als Quellaufwerke wählen kann. Hinter der Laufwerksnummer steht ein „+" oder ein „-" als Kennung für angeschlossen oder nicht vorhanden. Das gleiche steht auch in der nächsten Reihe, wo man die Ziellaufwerke auswählen kann. Danach kommt die Frage: Formatieren oder nicht. Und zum Schluß die Auswahl zwischen 1050 und einfacher Dichte. Ganz unten ohne Kasten stehen Status Request und Beginn zur Wahl. Nach dem Laden sind einige Standard Parameter schon gesetzt, sodaß man sofort beginnen kann. Alle Parameter werden am Rand in einer Statuszeile angezeigt.

Beim Kopieren wird wieder durch die Maus der Diskettenwechsel bestätigt. Tritt ein Fehler auf, kann man zwischen Wiederholung oder Abbruch wählen.

```
0 POKE 559,0
1000 REM * Binaer-File laden
1010 S=0:RESTORE 1100
1015 OPEN #1,8,0,"D:SCOPY.OBJ"
1016 PUT #1,255:PUT #1,255
1017 PUT #1,0:PUT #1,170
1018 PUT #1,0:PUT #1,183
1020 FOR A=43520 TO 46848:READ D:PUT #1,D:S=S+D:NEXT A
1030 IF S<>339993 THEN ? "DATEN-FEHLER!":STOP
1090 CLOSE #1:GRAPHICS 0:END
1100 DATA 76,48,176,174,0,2,172,1,2,142,237,182,140,238,182,120,216
1110 DATA 173,48,2,133,128,173,49,2,133,129,160,0,169,240,145,128
1120 DATA 169,192,141,14,212,169,82,141,0,2,169,170,141,1,2,169,58
1130 DATA 141,47,2,169,2,141,29,208,169,16,141,7,212,160,0,152,153
1140 DATA 0,20,153,0,21,200,208,247,88,169,1,141,111,2,96,120,72,138
1150 DATA 72,152,72,173,206,170,141,18,208,141,19,208,32,110,170,32
1160 DATA 170,170,104,168,104,170,104,88,64,174,0,211,138,41,1,208
1170 DATA 9,173,205,170,56,233,2,141,205,170,138,41,2,208,9,173,205
1180 DATA 170,24,105,2,141,205,170,138,41,4,208,9,173,204,170,56,233
1190 DATA 2,141,204,170,138,41,8,208,9,173,204,170,24,105,2,141,204
1200 DATA 170,96,173,204,170,141,0,208,24,105,8,141,1,208,172,205
1210 DATA 170,162,26,189,207,170,153,0,20,189,234,170,153,0,21,200
1220 DATA 202,16,240,96,112,100,15,0,0,0,0,0,0,0,0,0,0,7,15,31,31,15
1230 DATA 11,6,12,24,48,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,192,96,176
1240 DATA 208,224,224,96,96,48,0,0,0,0,0,0,0,0,0,0,168,173,7,228,72
1250 DATA 173,6,228,72,152,96,125,68,114,105,118,101,115,32,116,111
1260 DATA 32,99,104,111,111,115,101,32,97,115,32,115,111,117,114,99
1270 DATA 101,58,155,17,18,18,18,18,18,18,18,5,17,18,18,18,18,18,18
1280 DATA 5,17,18,18,18,18,18,5,17,18,18,18,18,18,18,5,155,124,32
```

```
1290 DATA 18,18,18,18,18,49,124,124,32,18,18,18,18,50,124,124,32,18
1300 DATA 18,18,51,124,124,32,18,18,18,18,52,124,155,26,18,18,18,18
1310 DATA 18,18,18,3,26,18,18,18,18,18,18,3,26,18,18,18,18,18,3,26
1320 DATA 18,18,18,18,18,18,3,155,68,114,105,118,101,115,32,116,111
1330 DATA 32,99,104,111,111,115,101,32,97,115,32,100,101,115,116,110
1340 DATA 46,58,155,17,18,18,18,18,18,18,18,5,17,18,18,18,18,18,18
1350 DATA 5,17,18,18,18,18,18,5,17,18,18,18,18,18,18,5,155,124,32
1360 DATA 18,18,18,18,18,49,124,124,32,18,18,18,18,50,124,124,32,18
1370 DATA 18,18,51,124,124,32,18,18,18,18,52,124,155,26,18,18,18,18
1380 DATA 18,18,18,3,26,18,18,18,18,18,18,3,26,18,18,18,18,18,3,26
1390 DATA 18,18,18,18,18,18,3,155,67,111,112,121,32,79,112,116,105
1400 DATA 111,110,115,58,155,17,18,18,18,18,18,18,18,18,18,18,18,18
1410 DATA 18,18,18,5,17,18,18,18,18,18,18,18,18,18,18,18,18,18,5,155
1420 DATA 124,70,111,114,109,97,116,32,68,101,115,116,110,46,32,32
1430 DATA 124,124,78,111,32,70,111,114,109,97,116,32,32,32,32,124
1440 DATA 155,26,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,3,26
1450 DATA 18,18,18,18,18,18,18,18,18,18,18,18,18,3,155,17,18,18,18
1460 DATA 18,18,18,18,18,18,18,18,18,18,18,18,5,17,18,18,18,18,18
1470 DATA 18,18,18,18,18,18,18,18,5,155,124,83,105,110,103,108,101
1480 DATA 32,68,101,110,115,105,116,121,32,124,124,49,48,53,48,32
1490 DATA 68,101,110,115,105,116,121,32,124,155,26,18,18,18,18,18
1500 DATA 18,18,18,18,18,18,18,18,18,18,3,26,18,18,18,18,18,18,18
1510 DATA 18,18,18,18,18,18,3,155,42,83,84,65,84,85,83,32,82,69,81
1520 DATA 85,69,83,84,42,32,32,42,83,69,67,84,79,82,32,67,79,80,89
1530 DATA 42,32,155,29,29,155,32,32,32,84,72,69,32,83,85,80,69,82
1540 DATA 32,83,69,67,84,79,82,32,67,79,80,73,69,82,155,32,32,32,40
1550 DATA 67,41,32,49,57,56,53,32,66,89,32,77,72,45,80,82,79,68,85
1560 DATA 67,84,73,79,78,83,155,32,32,32,87,82,73,84,84,69,78,32,66
1570 DATA 89,32,79,76,73,86,69,82,32,83,67,72,65,65,75,69,155,125
1580 DATA 17,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18
1590 DATA 18,18,18,18,18,18,18,18,18,18,18,18,5,155,124,32,68,105
1600 DATA 115,107,32,67,104,97,110,103,101,100,32,97,110,100,32,67
1610 DATA 111,110,116,105,110,117,101,32,32,32,32,124,155,26,18,18
1620 DATA 18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18
1630 DATA 18,18,18,18,18,18,18,18,18,3,155,125,17,18,18,18,18,18,18
1640 DATA 18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18
1650 DATA 18,18,18,18,18,5,155,124,32,69,114,114,111,114,32,102,111
1660 DATA 117,110,100,46,89,111,117,114,32,99,104,111,105,99,101,58
1670 DATA 32,32,32,32,32,124,155,26,18,18,18,18,18,18,18,18,18,18,18
1680 DATA 18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18
1690 DATA 18,3,155,17,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18
1700 DATA 18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,5,155,124,32
1710 DATA 69,114,114,111,114,46,82,101,116,114,121,32,79,112,101,114
1720 DATA 97,116,105,111,110,32,32,32,32,32,32,32,32,124,155,26,18
1730 DATA 18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18
1740 DATA 18,18,18,18,18,18,18,18,18,18,3,155,17,18,18,18,18,18,18
1750 DATA 18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18
1760 DATA 18,18,18,18,18,5,155,124,32,69,114,114,111,114,46,65,98
1770 DATA 111,114,116,32,79,112,101,114,97,116,105,111,110,32,32,32
1780 DATA 32,32,32,32,32,124,155,26,18,18,18,18,18,18,18,18,18,18,18
1790 DATA 18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18
1800 DATA 18,3,155,138,162,0,157,68,3,152,157,69,3,169,9,157,66,3
1810 DATA 169,255,157,72,3,169,0,141,47,2,32,86,228,169,58,141,47
```

```
1820 DATA 2,96,162,0,160,6,142,68,3,140,69,3,162,128,160,5,142,72
1830 DATA 3,140,66,3,162,0,32,86,228,96,162,16,160,171,32,235,174
1840 DATA 162,45,160,171,32,235,174,162,78,160,171,32,235,174,162
1850 DATA 111,160,171,32,235,174,162,144,160,171,32,235,174,162,172
1860 DATA 160,171,32,235,174,162,205,160,171,32,235,174,162,238,160
1870 DATA 171,32,235,174,162,15,160,172,32,235,174,162,29,160,172
1880 DATA 32,235,174,162,62,160,172,32,235,174,162,95,160,172,32,235
1890 DATA 174,162,128,160,172,32,235,174,162,161,160,172,32,235,174
1900 DATA 162,194,160,172,32,235,174,162,227,160,172,32,235,174,162
1910 DATA 4,160,173,32,235,174,162,7,160,173,32,235,174,162,34,160
1920 DATA 173,32,235,174,162,64,160,173,32,235,174,169,1,141,240,2
1930 DATA 133,9,162,48,160,176,134,12,132,13,169,15,162,0,160,114
1940 DATA 141,197,2,142,200,2,140,198,2,96,162,1,142,42,176,169,49
1950 DATA 141,0,3,24,172,42,176,192,5,240,71,140,1,3,169,83,141,2
1960 DATA 3,169,64,141,3,3,169,0,141,4,3,169,6,141,5,3,169,128,141
1970 DATA 8,3,169,0,141,9,3,32,83,228,16,2,48,12,174,42,176,202,169
1980 DATA 171,157,44,176,76,36,176,174,42,176,202,169,173,157,44,176
1990 DATA 76,36,176,238,42,176,76,215,175,0,96,46,46,46,46,32,210
2000 DATA 175,32,39,175,32,3,170,162,0,160,2,132,84,160,3,132,85,189
2010 DATA 44,176,32,5,171,162,1,160,2,132,84,160,12,132,85,189,44
2020 DATA 176,32,5,171,162,2,189,44,176,160,20,132,85,32,5,171,162
2030 DATA 3,189,44,176,160,27,132,85,32,5,171,162,0,189,44,176,160
2040 DATA 3,132,85,160,6,132,84,32,5,171,162,1,189,44,176,160,12,132
2050 DATA 85,32,5,171,162,2,189,44,176,160,20,132,85,32,5,171,162
2060 DATA 3,189,44,176,160,27,132,85,32,5,171,165,20,197,20,240,250
2070 DATA 32,239,182,32,245,182,32,115,178,32,76,178,32,190,176,76
2080 DATA 183,176,96,169,1,141,132,2,173,132,2,201,0,208,249,162,38
2090 DATA 236,205,170,240,56,232,224,51,208,246,162,68,236,205,170
2100 DATA 240,47,232,224,83,208,246,162,100,236,205,170,240,38,232
2110 DATA 224,115,208,246,162,124,236,205,170,240,29,232,224,141,208
2120 DATA 246,162,144,236,205,170,240,20,232,224,151,208,246,76,189
2130 DATA 176,76,24,177,76,95,177,76,166,177,76,207,177,76,247,177
2140 DATA 162,56,236,204,170,240,44,232,224,85,208,246,162,92,236
2150 DATA 204,170,240,37,232,224,116,208,246,162,124,236,204,170,240
2160 DATA 30,232,224,146,208,246,162,152,236,204,170,240,23,232,224
2170 DATA 181,208,246,76,189,176,162,1,76,27,178,162,2,76,27,178,162
2180 DATA 3,76,27,178,162,4,76,27,178,162,56,236,204,170,240,44,232
2190 DATA 224,85,208,246,162,92,236,204,170,240,37,232,224,116,208
2200 DATA 246,162,124,236,204,170,240,30,232,224,147,208,246,162,152
2210 DATA 236,204,170,240,23,232,224,181,208,246,76,189,176,162,1
2220 DATA 76,48,178,162,2,76,48,178,162,3,76,48,178,162,4,76,48,178
2230 DATA 162,56,236,204,170,240,17,232,224,117,208,246,162,124,236
2240 DATA 204,170,240,14,232,224,181,208,246,173,69,178,141,72,178
2250 DATA 76,76,178,169,0,141,72,178,76,76,178,162,56,236,204,170
2260 DATA 240,17,232,224,117,208,246,162,124,236,204,170,240,13,232
2270 DATA 224,181,208,246,162,33,142,69,178,76,115,178,162,34,142
2280 DATA 69,178,76,115,178,162,56,236,204,170,240,20,232,224,117
2290 DATA 208,246,162,124,236,204,170,240,14,232,224,181,208,246,76
2300 DATA 189,176,76,48,176,76,189,176,76,73,178,142,70,178,138,24
2310 DATA 105,48,162,0,160,38,134,84,132,85,32,5,171,76,189,176,142
2320 DATA 71,178,138,24,105,48,162,4,160,38,134,84,132,85,32,5,171
2330 DATA 76,189,176,33,1,1,0,76,103,180,173,72,178,201,0,240,16,169
2340 DATA 89,162,10,160,38,134,84,132,85,32,5,171,76,189,176,169,78
```

```
2350 DATA 162,10,160,38,134,84,132,85,32,5,171,76,189,176,173,69,178
2360 DATA 201,33,240,16,169,68,162,13,160,38,134,84,132,85,32,5,171
2370 DATA 76,189,176,169,83,162,13,160,38,134,84,132,85,32,5,171,76
2380 DATA 189,176,169,64,141,14,212,169,49,141,0,3,173,70,178,141
2390 DATA 1,3,169,82,141,2,3,169,64,141,3,3,173,69,182,141,10,3,173
2400 DATA 70,182,141,11,3,173,71,182,141,4,3,173,72,182,141,5,3,169
2410 DATA 128,141,8,3,169,0,141,9,3,32,83,228,48,35,24,173,69,182
2420 DATA 105,1,141,69,182,173,70,182,105,0,141,70,182,24,173,71,182
2430 DATA 105,128,141,71,182,173,72,182,105,0,141,72,182,96,162,193
2440 DATA 160,173,32,235,174,162,227,160,173,32,235,174,162,4,160
2450 DATA 174,32,235,174,162,37,160,174,32,235,174,162,70,160,174
2460 DATA 32,235,174,162,103,160,174,32,235,174,162,136,160,174,32
2470 DATA 235,174,162,169,160,174,32,235,174,162,202,160,174,32,235
2480 DATA 174,32,11,182,76,154,178,169,64,141,14,212,169,49,141,0
2490 DATA 3,173,71,178,141,1,3,169,80,141,2,3,169,128,141,3,3,173
2500 DATA 69,182,141,10,3,173,70,182,141,11,3,173,71,182,141,4,3,173
2510 DATA 72,182,141,5,3,169,128,141,8,3,169,0,141,9,3,32,89,228,48
2520 DATA 35,24,173,69,182,105,1,141,69,182,173,70,182,105,0,141,70
2530 DATA 182,24,173,71,182,105,128,141,71,182,173,72,182,105,0,141
2540 DATA 72,182,96,162,193,160,173,32,235,174,162,227,160,173,32
2550 DATA 235,174,162,4,160,174,32,235,174,162,37,160,174,32,235,174
2560 DATA 162,70,160,174,32,235,174,162,103,160,174,32,235,174,162
2570 DATA 136,160,174,32,235,174,162,169,160,174,32,235,174,162,202
2580 DATA 160,174,32,235,174,32,11,182,76,67,179,169,64,141,14,212
2590 DATA 169,49,141,0,3,173,71,178,141,1,3,173,69,178,141,2,3,169
2600 DATA 34,141,3,3,169,255,141,6,3,141,7,3,32,89,228,48,7,165,20
2610 DATA 197,20,240,250,96,162,193,160,173,32,235,174,162,227,160
2620 DATA 173,32,235,174,162,4,160,174,32,235,174,162,37,160,174,32
2630 DATA 235,174,162,70,160,174,32,235,174,162,103,160,174,32,235
2640 DATA 174,162,136,160,174,32,235,174,162,169,160,174,32,235,174
2650 DATA 162,202,160,174,32,235,174,32,11,182,76,236,179,96,32,236
2660 DATA 179,76,153,180,169,1,141,69,182,169,0,141,70,182,169,0,141
2670 DATA 71,182,169,22,141,72,182,32,73,182,32,154,178,173,69,182
2680 DATA 201,41,208,246,173,70,182,201,1,208,239,32,135,182,173,72
2690 DATA 178,201,0,208,200,169,1,141,69,182,169,0,141,70,182,169
2700 DATA 0,141,71,182,169,22,141,72,182,32,67,179,173,69,182,201
2710 DATA 41,208,246,173,70,182,201,1,208,239,32,73,182,169,41,141
2720 DATA 69,182,169,1,141,70,182,169,0,141,71,182,169,22,141,72,182
2730 DATA 32,154,178,173,69,182,201,81,208,246,173,70,182,201,2,208
2740 DATA 239,169,41,141,69,182,169,1,141,70,182,169,0,141,71,182
2750 DATA 169,22,141,72,182,32,135,182,32,67,179,173,69,182,201,81
2760 DATA 208,246,173,70,182,201,2,208,239,32,73,182,169,81,141,69
2770 DATA 182,169,2,141,70,182,169,0,141,71,182,169,22,141,72,182
2780 DATA 32,154,178,173,69,182,201,209,208,246,173,70,182,201,2,208
2790 DATA 239,169,81,141,69,182,169,2,141,70,182,169,0,141,71,182
2800 DATA 169,22,141,72,182,32,135,182,32,67,179,173,69,182,201,209
2810 DATA 208,246,173,70,182,201,2,208,239,173,69,178,201,34,240,3
2820 DATA 76,51,176,169,209,141,69,182,169,2,141,70,182,169,0,141
2830 DATA 71,182,169,22,141,72,182,32,73,182,32,154,178,173,69,182
2840 DATA 201,249,208,246,173,70,182,201,3,208,239,169,209,141,69
2850 DATA 182,169,2,141,70,182,169,0,141,71,182,169,22,141,72,182
2860 DATA 32,135,182,32,67,179,173,69,182,201,249,208,246,173,70,182
2870 DATA 201,3,208,239,32,73,182,169,249,141,69,182,169,3,141,70
```

```
2880 DATA 182,169,0,141,71,182,169,22,141,72,182,32,154,178,173,69
2890 DATA 182,201,17,208,246,173,70,182,201,4,208,239,169,249,141
2900 DATA 69,182,169,3,141,70,182,169,0,141,71,182,169,22,141,72,182
2910 DATA 32,135,182,32,67,179,173,69,182,201,17,208,246,173,70,182
2920 DATA 201,4,208,239,76,51,176,169,192,141,14,212,173,132,2,201
2930 DATA 0,208,249,162,34,236,205,170,240,32,232,224,43,208,246,162
2940 DATA 44,236,205,170,240,23,232,224,66,208,246,162,69,236,205
2950 DATA 170,240,12,232,224,91,208,246,76,16,182,104,104,96,96,76
2960 DATA 51,176,0,0,0,0,32,3,170,162,93,160,173,32,235,174,162,127
2970 DATA 160,173,32,235,174,162,160,160,173,32,235,174,162,197,160
2980 DATA 182,32,235,174,173,70,178,205,71,178,208,22,173,132,2,201
2990 DATA 0,208,249,162,28,236,205,170,240,8,232,224,51,208,246,76
3000 DATA 112,182,96,32,3,170,162,93,160,173,32,235,174,162,127,160
3010 DATA 173,32,235,174,162,160,160,173,32,235,174,162,217,160,182
3020 DATA 32,235,174,173,70,178,205,71,178,208,22,173,132,2,201,0
3030 DATA 208,249,162,29,236,205,170,240,8,232,224,51,208,246,76,174
3040 DATA 182,96,73,78,83,69,82,84,32,83,79,85,82,67,69,32,68,73,83
3050 DATA 75,46,155,73,78,83,69,82,84,32,68,69,83,84,78,46,32,68,73
3060 DATA 83,75,46,155,194,192,174,70,178,76,27,178,174,71,178,76
3070 DATA 48,178,0,255,255,0,0,0
```


! ACHTEN SIE AUF UNSERE SUPER-PROGRAMM-ANGEBOTE-Seite!

Kleinanzeigen

Liebe Leser,

wollen Sie eine Kleinanzeige aufgeben? Suchen Sie einen gebrauchten Home-Computer, Software, Hardware, Kontakte zu Computer-Clubs etc.?

Dann nutzen Sie jetzt den Kleinanzeigenservice der „CPU": Die „CPU" ist ein kompetentes Anwender-Magazin.

Sie erreichen durch uns eine Vielzahl von Lesern im In- und Ausland.

Und so wird's gemacht:

Kleinanzeigen-Coupon ausfüllen (alle Zeilen können ausgefüllt werden) und **einsenden an**

Tronic-Verlag
Postfach 41
3444 Wehretal 1

Also, schnell ausfüllen und einwerfen in den nächsten Postkasten. Dann ist mit Sicherheit Ihre Kleinanzeige dabei.

- CPU - Basar

Kleinanzeigen

- CPU - Basar

Bitte veröffentlichen Sie ab nächstmöglicher Ausgabe unter der Rubrik:

☐ SOFTWARE ☐ HARDWARE ☐ VERSCHIEDENES

Mein Computersystem: _____

folgenden Text:

Bitte jeweils 28 Buchstaben pro Zeile! Bitte Absender nicht vergessen!

Preis für Kleinanzeigen: 10 DM je Kleinanzeige
(keine gewerbliche (Nutzung) ☐ Scheck ☐ Bar
☐ kostenlose Kleinanzeige für unsere Juroren (siehe S. 91)!

Name/Vorname:

Straße, Nr.: PLZ/Ort:

Datum, Unterschrift

Ich versichere hiermit, daß die angebotenen Programme/Beiträge frei von Rechten Dritter sind!

Mit 20-Bytes Maschinenprogramm ein vollständiger Zeilenlöscher

Zeilenkill für den ZX-Spectrum 16/48K

In der Kürze liegt die Würze. Mit nur 20 Bytes Maschinencode, der zudem noch an keinerlei feste Laufadresse gebunden ist, gibt es einen vollwertigen Zeilenlöscher. Damit kann das Programm bei Bedarf jederzeit nachgeladen werden. Auf Menüsteuerung wurde verzichtet, denn damit würde das Programm nur unnötig vergrößert. Was für einen Zweck hat das beste Hilfsprogramm, wenn es mehr Speicherplatz benötigt als erforderlich. Die Handhabung des Programms ist dadurch keinesfalls schwierig geworden.

Mit dem Programm lassen sich beliebig viele Zeilen aus einem Basicprogramm löschen. Der Basicprogrammierer kennt den Zeitaufwand, der von Nöten ist, wenn es darum geht eine größere Anzahl von Zeilen zu entfernen. Nun kann uns aber das Programm helfen.

Zur Eingabe:
Wenn Sie keinen Assembler zur Verfügung haben, nehmen Sie das kurze Eingabeprogramm. Nach der Eingabe Starten Sie mit ‚Run' und ‚SAVE'n' es mit ‚GOTO 100' ab. Es wird nur der Maschinencode gespeichert, dies ist ja auch das Nutzprogramm.
Durch ‚LOAD"Zeilenkill" CODE XXXX kann es nun an jeder freien Speicherstelle geladen werden. Es kann auch im Druckspeicher verbleiben, der die meiste Zeit nicht benötigt wird, wenn doch, kann es schnell geladen werden.
Nun noch ein Beispiel:
LIST (erste zu löschende Zeile)
RANDOMIZE (erste verbleibende Zeile)
RANDOMIZE USR (Adresse des Maschinencode)
Alle drei Eingaben werden mit ‚ENTER' abgeschlossen. Die letzten beiden Befehlsfolgen können durch ‚:' getrennt zusammen eingegeben werden.

Also:
LIST 5
RANDOMIZE 2890
RANDOMIZE USR 23296
würde alle Zeilen von 5 bis 2889 löschen.

Gerd Raskopp

```
10 REM ************************
20 REM    Gerd Raskopp
30 REM    Aumunder Feldstr.70
40 REM     2820 Bremen 70
42:
43 REM  Programm "Zeilenkill"
44:
46 REM  Spectrum 16K und 48K
50 REM ************************
```

```
  60 FOR n=23296 TO 23315
  70 READ x: POKE n,x: NEXT n
  80 DATA 229,42,73,92,205,110,2
5,229,42,118,92,205,110,25,209,2
05,229,25,225,201
  90 STOP
 100 SAVE "Zeilenkill"CODE 23296
,20
```

CPC-464 * CPC-464 * CPC-464 → **Data-Generator** ← CPC-464 * CPC-464 * CPC-464

Hiermit lassen sich auf einfache Art und Weise Maschinenroutinen in Datazeilen überführen, so daß solche Programme auch an User weitergegeben werden können, die keinen Assembler besitzen.

Nachdem man das Listing abgetippt hat, sollte man das Programm zunächst saven, bevor es das erste Mal gestartet wird. Die Maschinenroutine löscht das Basic-Programm!

Nach dem Start (das umzuwandelnde Programm muß sich natürlich im Speicher befinden) wird zunächst nach der Start- und der Endadresse des umzuwandelnden Programms gefragt. Anschließend ist die Zahl der DATAs, die man pro Basic-Zeile wünscht, einzugeben. Es kann zwischen 1 und 79 gewählt werden. Gibt man nur ‚ENTER' oder einen falschen Wert ein, so werden automatisch 20 DATAs pro Zeile generiert. Hat man die letzte Eingabe mit ‚ENTER' abgeschlossen, so meldet sich der Rechner fast augenblicklich mit:

'SYNTAX ERROR IN 1000' und zeigt die 1. DATA-Zeile an.

Diese Fehlermeldung ist ohne Bedeutung! Sie kommt deshalb zustande, weil das aufrufende Programm gelöscht wurde.

Die DATA-Zeilen beginnen mit Zeilennummer 1000 und lassen sich mit RENUM beliebig verschieben. Das Programm ist sehr schnell, 1000 DATAs werden in weniger als einer Sekunde geschrieben.

Das Lesen der DATAs muß über READ a$ geschehen, anschließend wird der String durch a=VAL("&"+a$) in eine Zahlvariable umgewandelt. (Man kann sich übrigens eine Menge Tipperei ersparen, wenn man bei Listings, die DATAs mit vorangestellten "&" enthalten, dieses jeweils wegläßt und stattdessen die Lese-Routine in die obengenannte Form bringt.)

Mit den Eingaben &8000, &80EC und 20 werden die DATA-Zeilen des Programms erzeugt. Das vorletzte DATA weicht vom Listing ab, es enthält das Low-Byte der Zeilennummer.

Wolfgang Bönigk

CPU sucht Programmbeiträg!

Für die Rubriken „User", „Utility", „Software- und Hardwaretest" suchen wir Beiträge.

Honorar: 120,- DM pro veröffentlichte Seite!

BASIC-LOADER

```
100 GOSUB 210 : CALL &8000,@s%,@e%,@d%
110 '
120 '************************************************
130 '*****                                      *****
140 '*****         D A T A G E N E R A T O R    *****
150 '*****           fuer Schneider CPC 464     *****
160 '*****                                      *****
170 '*****    (C)  Wolfgang Boenigk   Osnabrueck *****
180 '*****                                      *****
190 '************************************************
200 '
210 MODE 2
220 MEMORY &7FFF : adr=&8000
230 WHILE a$<>"ende"
240    READ a$
250    a=VAL("&"+a$)
260    POKE adr+i,a
270    sum=sum+a : i=i+1
280 WEND
290 IF sum<>29972 THEN PRINT"Fehler in DATAs !":END
300 '
310 LOCATE 5,8:INPUT "Startadresse    : ";s%
320 LOCATE 5,10:INPUT "Endadresse      : ";e%
330 LOCATE 5,12:INPUT "DATAs pro Zeile : ";d%
340 RETURN
350 '
1000 DATA FE,03,C0,DD,66,01,DD,6E,00,7E,FE,50,38,04,3E,14,18,04,FE,00
1001 DATA 28,F8,32,EF,80,CD,D8,80,DD,6E,02,DD,66,03,5E,23,56,ED,53,ED
1002 DATA 80,DD,6E,04,DD,66,05,5E,23,56,D5,DF,E8,80,FD,E1,DD,21,70,01
1003 DATA 3A,F0,80,DD,77,00,DD,E5,DD,23,DD,36,00,00,DD,23,2A,EB,80,23
1004 DATA 22,EB,80,DD,75,00,DD,23,DD,74,00,DD,23,DD,36,00,8C,DD,23,DD
1005 DATA 36,00,20,DD,23,3A,EF,80,47,C5,CD,B2,80,DD,36,00,2C,DD,23,C1
1006 DATA FD,E5,D1,2A,ED,80,AF,ED,52,38,0A,10,E8,DD,36,FF,00,E1,C3,3C
1007 DATA 80,05,3A,EF,80,90,CD,D8,80,E1,77,DD,36,FF,00,DD,23,DD,23,DD
1008 DATA 23,DD,22,83,AE,DD,22,85,AE,DD,22,87,AE,DD,22,89,AE,C9,FD,6E
1009 DATA 00,FD,23,26,00,11,10,00,CD,C1,BD,CD,C7,80,6B,CD,C7,80,C9,7D
1010 DATA FE,0A,38,08,C6,37,DD,77,00,DD,23,C9,C6,30,18,F6,6F,26,00,11
1011 DATA 03,00,CD,BE,BD,7D,C6,06,32,F0,80,C9,3E,C1,FD,E7,03,ende
```

Assemblerlisting

```
10                          ORG   8000H
11                          LOAD  8000H
12              PROEND:     EQU   0AE83H        ;Programmende
13              VARANF:     EQU   0AE85H        ;Variablenstart
14              ARRSTA:     EQU   0AE87H        ;Stringstart
15              ARREND:     EQU   0AE89H        ;Stringende
16              MULT:       EQU   0BDBEH        ;Multiplikation
17              DIV:        EQU   0BDC1H        ;Division
18              ;
19              ;** Eingabe pruefen **
20 8000 FE03                CP    3
21 8002 C0                  RET   NZ
```

```
22 8003 DD6601              LD   H,(IX+1)
23 8006 DD6E00              LD   L,(IX+0)
24 8009 7E                  LD   A,(HL)
25 800A FE50                CP   80
26 800C 3804                JR   C,LANG1
27 800E 3E14      LANG2:    LD   A,20
28 8010 1804                JR   INIT
29 8012 FE00      LANG1:    CP   0
30 8014 28F8                JR   Z,LANG2
31                ;
32                ;** Initialisierung **
33 8016 32EF80    INIT:     LD   (DATAS),A        ;DATAs pro Zeile
34 8019 CDD880              CALL LAENGE
35 801C DD6E02              LD   L,(IX+2)
36 801F DD6603              LD   H,(IX+3)
37 8022 5E                  LD   E,(HL)
38 8023 23                  INC  HL
39 8024 56                  LD   D,(HL)
40 8025 ED53ED80            LD   (BIS),DE         ;Endadresse
41 8029 DD6E04              LD   L,(IX+4)
42 802C DD6605              LD   H,(IX+5)
43 802F 5E                  LD   E,(HL)
44 8030 23                  INC  HL
45 8031 56                  LD   D,(HL)
46 8032 D5                  PUSH DE               ;Startadresse
47 8033 DF                  RST  18H              ;Basic-Programm loeschen
48 8034 E880                DEFW FARADR
49 8036 FDE1                POP  IY
50 8038 DD217001            LD   IX,170H          ;Basicstart
51                ;
52                ;** Zeilenanfang schreiben **
53 803C 3AF080    NEXTZE:   LD   A,(BYTES)
54 803F DD7700              LD   (IX+0),A         ;Laenge in Bytes
55 8042 DDE5                PUSH IX               ;Adr.v.Zeilenlaenge
56 8044 DD23                INC  IX
57 8046 DD360000            LD   (IX+0),0
58 804A DD23                INC  IX
59 804C 2AEB80              LD   HL,(ZEINUM)      ;Zeilennummer
60 804F 23                  INC  HL               ;erhoehen
61 8050 22EB80              LD   (ZEINUM),HL
62 8053 DD7500              LD   (IX+0),L         ;Low-Byte schreiben
63 8056 DD23                INC  IX
64 8058 DD7400              LD   (IX+0),H         ;High-Byte schreiben
65 805B DD23                INC  IX
66 805D DD36008C            LD   (IX+0),8CH       ;Token "DATA"
67 8061 DD23                INC  IX
68 8063 DD360020            LD   (IX+0),20H       ;Blank
69 8067 DD23                INC  IX
70 8069 3AEF80              LD   A,(DATAS)
71 806C 47                  LD   B,A
72                ;
73                ;** DATAs schreiben **
74 806D C5        LOOP:     PUSH BC
75 806E CDB280              CALL LES
76 8071 DD36002C            LD   (IX+0),","
77 8075 DD23                INC  IX
78 8077 C1                  POP  BC
79 8078 FDE5                PUSH IY               ;letzte gelesene Adresse
80 807A D1                  POP  DE
81 807B 2AED80              LD   HL,(BIS)
82 807E AF                  XOR  A
83 807F ED52                SBC  HL,DE            ;fertig?
```

```
 84 8081 380A                JR   C,FERTIG
 85 8083 10E8                DJNZ LOOP
 86 8085 DD36FF00            LD   (IX-1),0          ;Token 'Zeilenende'
 87 8089 E1                  POP  HL
 88 808A C33C80              JP   NEXTZE            ;naechste Zeile
 89                ;
 90                ;*letzte Zeilenlaenge korrigieren*
 91 808D 05        FERTIG:   DEC  B
 92 808E 3AEF80              LD   A,(DATAS)
 93 8091 90                  SUB  B
 94 8092 CDD880              CALL LAENGE
 95 8095 E1                  POP  HL                ;Adr.letzter Zeilenlaenge
 96 8096 77                  LD   (HL),A
 97 8097 DD36FF00            LD   (IX-1),0
 98                ;
 99                ;** Programm- und Variablen- **
100                ;**     pointer setzen        **
101 809B DD23               INC  IX
102 809D DD23               INC  IX
103 809F DD23               INC  IX
104 80A1 DD2283AE           LD   (PROEND),IX
105 80A5 DD2285AE           LD   (VARANF),IX
106 80A9 DD2287AE           LD   (ARRSTA),IX
107 80AD DD2289AE           LD   (ARREND),IX
108 80B1 C9                 RET
109               ;
110               ;** Byte lesen und zerlegen **
111 80B2 FD6E00   LES:      LD   L,(IY+0)
112 80B5 FD23               INC  IY
113 80B7 2600               LD   H,0
114 80B9 111000             LD   DE,16
115 80BC CDC1BD             CALL DIV
116 80BF CDC780             CALL UMR
117 80C2 6B                 LD   L,E
118               ;
119               ;** Umwandlung nach ASCII **
120 80C3 CDC780             CALL UMR
121 80C6 C9                    RET
122 80C7 7D         UMR:       LD   A,L
123 80C8 FE0A                  CP   10
124 80CA 3808                  JR   C,ZAHL
125 80CC C637                  ADD  A,37H             ;wenn Buchstabe
126                ;
127                ;** ASCII-Zeichen schreiben **
128 80CE DD7700    POKE:       LD   (IX+0),A
129 80D1 DD23                  INC  IX
130 80D3 C9                    RET
131 80D4 C630      ZAHL:       ADD  A,30H             ;wenn Zahl
132 80D6 18F6                  JR   POKE
133                ;
134                ;** Laenge der Zeile berechnen **
135 80D8 6F        LAENGE:     LD   L,A
136 80D9 2600                  LD   H,0
137 80DB 110300                LD   DE,3
138 80DE CDBEBD                CALL MULT
139 80E1 7D                    LD   A,L
140 80E2 C606                  ADD  A,6
141 80E4 32F080                LD   (BYTES),A
142 80E7 C9                    RET
```

In der nächsten CPU:

Für den Schneider-Computer!

- **Maschinensprache-Programm zum Gestalten von farbigen Grafiken; Grafiken können in eigene Programme eingebunden werden.**
- **Checksummen für die veröffentlichten CPC-Programme: Prüfsummen verhindern jetzt Tippfehler bei der Eingabe!**

```
143 80E8 3EC1      FARADR:     DEFW 0C13EH          ;Einsprungadr.
144 80EA FD                    DEFB 0FDH            ;in Basic-ROM
145 80EB E703      ZEINUM:     DEFW 999             ;Zeilennummer
146                BIS:        DEFS 2
147                DATAS:      DEFS 1
148                BYTES:      DEFS 1
149                            END
```

Schneider CPC-464 - Graphic TOOLKIT

Grafik Toolkit ist eine Erweiterung des Locomotive BASIC um sechs weitere Befehle, die die Grafikfähigkeiten des Schneidercomputers CPC-464 noch weiter verbessern.

Das eingesandte Programmpaket besteht aus drei Einzelprogrammen:

1. TOOLKIT: Dies ist das Ladeprogramm, welches den Zeichensatz nach unten verschiebt, den BASIC-Speicher begrenzt und das nachgeladene Maschinencodeprogramm in den BASIC-Interpreter des CPC-464 einbindet.
2. CREATOR: Er erzeugt den Maschinencode und speichert ihn unter dem Namen „TOOL" ab.
3. DEMO: Ist ein Demonstrationsprogramm, in dem die sechs neuen Befehle vorgestellt werden.

Die Befehle selbst sind keine RSX-Kommendos, sondern lassen sich genauso unkompliziert wie die herkömmlichen Befehle des CPC-464 einsetzen. Die BASIC-Erweiterung ist zudem noch vollständig Diskettenkompatibel, da der unter AMSDOS reservierte Speicherbereich auch weiterhin unangetastet bleibt.

Einzelbeschreibung der Befehle:

- BOX x,y: Zeichnet ein Rechteck, ausgehend von der aktuellen Position des Grafikcursors, mit der Ausdehnung x in der waagrechten und y in senkrechter Richtung.
- CIRCLE x,y: Zeichnet einen Kreis bzw. eine Elipse mit dem Grafikcursor als Mittelpunkt und den Radien x und y.
- COLOR m,n: Legt Graphic Pen und Graphic Paper fest.
- MAGNIFY a,x,y: Gibt den Inhalt des Strings a, ausgehend von der Position des Grafikcursors, um das x-fache in waagrechter und das y-fache in senkrechter Richtung vergrößert aus.
- PAINT m: Füllt eine umschlossene Fläche, bzw. bis an die Grenzen des Grafikfensters, mit der Farbe m ausgehend von Grafikcursor.
- SCREEN v: Schaltet die Bildschirmausgabe ab (v=Ø), bzw. wieder an (V=1).

Anweisung zum Erstellen einer schlüsselfertigen Version anhand der Listings:

- Listing 1 eintippen und abspeichern mit dem Befehl SAVE"TOOLKIT
- Listing 2 zur Sicherheit selbst irgendwo abspeichern ("CREATOR"), darauf starten und den erzeugten Maschinencode direkt hinter "TOOLKIT" abspeichern.
- Listing 3 erst eintippen und anwenden nachdem das Programm "TOOLKIT" wieder geladen und abgelaufen it.

Zur Anwendung RUN"TOOLKIT oder ggfls. CTRL+ENTER eingeben, erst danach kann man mit den neuen Befehlen arbeiten!

Thomas Trippel

Toolkit-Loader

```
100 SYMBOL AFTER 256:MEMORY 41906
110 POKE &AE7D,178:POKE &AE7E,163
120 POKE &B296,178:POKE &B297,163
130 CLEAR:SYMBOL AFTER 240
140 LOAD"!tool",41907
150 FOR n=1 TO 3:READ a,b
160 POKE a,&C3:POKE a+1,b AND 255
170 POKE a+2,INT(b/256)AND 255
180 NEXT:BORDER 2:MODE 1:LOCATE 2,2
190 PRINT"Graphic TOOLKIT 1.0   ";
200 PRINT"41408 Bytes free",,:NEW
210 DATA &ac07,&a61d
220 DATA &ac13,&a6b5
230 DATA &ac16,&a675
```

Toolkit-Generator

```
10 '*******************
20 '*                 *
30 '* Graphic TOOLKIT *
40 '*     (C)1985     *
50 '* by Thomas Fippl *
60 '* 6336 Solms/Lahn *
70 '*                 *
80 '*******************
90 '
100 MEMORY 39999:CLS:FOR z=0 TO 52
110 PRINT CHR$(30)"Zeile:"200+z;
120 READ x$,p
130 FOR l=1 TO LEN(x$)STEP 2
140 m=VAL("&"+MID$(x$,l,2))
150 POKE 39999+z*16+(l+1)/2,m
160 p=p XOR m:NEXT
170 IF p THEN PRINT"Error":STOP
180 NEXT:PRINT"ok"
190 SAVE"tool",b,40000,841
200 DATA cd9fced5cd37dd2ccd27c3e378a7281f,96
201 DATA 1c1ccb3b147b5a57d5dde1e5cde2a5e1,245
202 DATA e723e5cde6a32aaaac22a4ace110f1e1,78
203 DATA c306a62aa6ac22a8accdd3120e087e23,50
204 DATA e5cd04a42aacac22a8ace10d20f0c30e,117
205 DATA a62aa4ac22aaacc5060817c5cd22a416,228
206 DATA 00dd5d2aaaac1922aaacc110edc1c92a,143
207 DATA a8ac22acacdde5c13013c5f5ed5baaac,124
208 DATA cdf415160059626acd3618f1c12aacac,146
209 DATA 2b2b22acac10e1c9cd4bc2e5cde2a5cd,252
210 DATA f617cd90a4300bcd79a4010100cda4a4,254
211 DATA 18f0ed5ba4ac01ffffcda4a4cd90a430,83
212 DATA 68cd79a418f02b2bcd90a438f92aa6ac,136
213 DATA 2323cd90a438f92aa6acc30ea6e5d5cd,142
214 DATA fc16300acda90b7ea137cc1d183fd1e1,145
215 DATA c9eb3ac8b1fe012804300309090909eb,106
216 DATA c9cdf0a5c5e3ebe5d5e5210000e5cd36,118
217 DATA 18d1e1d5cd3618d1e1cdd437ebe5cd36,245
218 DATA 18d1e1cdd437cd3618e1c306a6cdf0a5,255
219 DATA c5e3ebe5d5e5d5e5cde5a5d119222eb3,210
```

```
220 DATA eb11b5accd65a5e111b0accd65a5061d,20
221 DATA 213ba65e235623cd73a510f7c12aa6ac,169
222 DATA a7ed4222a6ac061d2174a67eed44572b,99
223 DATA 5e2bcd73a510f4c12aa4aca7ed4222a4,231
224 DATA ac061d213ba67eed445f237eed445723,3
225 DATA cd73a510f1c12aa6ac0922a6ac061e21,3
226 DATA 74a6562b7eed445f2bcd73a510f4e1c3,209
227 DATA 06a6cd292e116ea5c39e340000007f88,46
228 DATA e5c5ebe511aaacd56c2600cd292e11b5,200
229 DATA accd1534cd692ed1e32600cd292e11b0,181
230 DATA accd1534cd692eed5ba4ac19e3ed5ba6,158
231 DATA ac19d1cd3c18c1e1c93e02cdfbc1e521,107
232 DATA feb1a7200ccdddbc11c2a5e1c322bd14,183
233 DATA 14cddabcaf32fbb111eab1e1c325bdcd,85
234 DATA 4bc2cddebbcd55ddd0cd4bc2c3e4bbcd,161
235 DATA fba5cdfc15ed53a4ac22a6acc9cd86ce,168
236 DATA d5cd55dddc86cec1f3d9cb91cbd9ed49,3
237 DATA d9fbc9f3d9cbd1cb9918f3cd421ad0fe,217
238 DATA fcc0cd06a6cd43c418ded6bafe0bd0c1,85
239 DATA eb212da6856f8cc3b9ddb4a4e0a4d2a5,163
240 DATA b3a34ba4aca501ff0dff1bfe28fc35f9,7
241 DATA 42f64ff35bee68e974e380dd8bd696ce,243
242 DATA a0c6abbeb4b4beabc6a0ce96d68bdd7f,167
243 DATA e374e968ee5bf34ff642f935fc28fe1b,86
244 DATA ff0de511dba6eb0edd1acd8affbe280c,123
245 DATA 7e231730fb0c7ea720efe1c923137ee6,27
246 DATA 7f471acd8affb82015be28f0eb237ecd,10
247 DATA 7bff3809f1f1f179d1c1c383dfebd1d5,93
248 DATA 18cefedd3804fee33802a7c9333323f5,86
249 DATA e521dba6d6dd280ac5477e231730fb10,131
250 DATA f9c1cd1ae2c337e2424fd8434952434c,153
251 DATA c5434f4c4fd24d41474e4946d9504149,147
252 DATA 4ed45343524545ce00,22
```

Demo-Programm

```
1000 '
1010 'Initialisierung
1020 '
1030 CLEAR
1040 DEFINT f,m,n,r,x,y
1050 DEFSTR c
1060 FOR n=0 TO 5
1070 READ m
1080 INK n,m
1090 NEXT
1100 DATA 20,0,4,8,14,17
1110 BORDER 18
1120 COLOR 1
1130 '
1140 'Haupt-Schleife
1150 '
1160 FOR n=1 TO 6
1170 ON n GOSUB 1210,1360,1470,1620,1740,1990
1180 GOSUB 2380
1190 NEXT
1200 END
1210 '
1220 '**** BOX ****
1230 '
1240 cmd="BOX"
1250 GOSUB 2250
1260 MODE 1
1270 MOVE 320,200
1280 FOR m=0 TO 150 STEP 10
1290 mm=m*1.25
1300 BOX mm,m
1310 BOX-m,mm
1320 BOX-mm,-m
1330 BOX m,-mm
1340 NEXT
1350 RETURN
1360 '
1370 '**** CIRCLE ****
1380 '
1390 cmd="CIRCLE"
1400 GOSUB 2250
1410 MODE 1
1420 FOR m=0 TO 190 STEP 10
1430 MOVE 320,200
1440 CIRCLE m,190-m
1450 NEXT
1460 RETURN
1470 '
1480 '**** COLOR ****
1490 '
1500 cmd="COLOR"
1510 GOSUB 2250
1520 MODE 1
1530 TAG
1540 FOR m=0 TO 3
1550 MOVE 10+m*25,250-m*20
1560 COLOR m+2,m
1570 PRINT"Graphic Pen:"(m+2)AND 3;
1580 PRINT"/ Graphic Paper:"m;
1590 NEXT
1600 TAGOFF
1610 RETURN
1620 '
1630 '**** MAGNIFY ****
1640 '
1650 cmd="MAGNIFY"
1660 GOSUB 2250
1670 MODE 0
1680 FOR m=0 TO 5
1690 COLOR 12-m
1700 MOVE 4+m*4,280-m*2
1710 MAGNIFY"CPC 464",11,20
1720 NEXT
1730 RETURN
1740 '
1750 '**** PAINT ****
1760 '
1770 cmd="PAINT"
1780 GOSUB 2250
1790 MODE 0
1800 FOR m=50 TO 350 STEP 100
1810 MOVE 0,m
1820 GOSUB 1910
1830 NEXT
1840 :
1850 FOR m=0 TO 3
1860 MOVE 320,m*100
1870 PAINT 2+m
1880 NEXT
1890 RETURN
1900 :
1910 y=0
1920 FOR x=8 TO 640 STEP 8
1930 r=INT(RND*11)-5
1940 y=y+r
1950 IF ABS(y)>39 THEN y=y-r
1960 DRAW x,m+y,1
1970 NEXT
1980 RETURN
1990 '
2000 '**** SCREEN ****
2010 '
2020 cmd="SCREEN"
2030 GOSUB 2250
2040 MODE 1
2050 SCREEN 0
2060 TAG
2070 COLOR 1,0
2080 MOVE 40,40
2090 DRAWR 560,0,1
2100 FOR x=1 TO 10
2110 MOVE x*56-10,30
2120 PRINT USING"##";x;
2130 FOR y=0 TO INT(RND*300)STEP 2
2140 MOVE x*56-16,50+y
2150 DRAWR 40,0,2
```

```
2160 DRAWR 8,8,3
2170 NEXT
2180 FOR m=XPOS-8 TO XPOS-48 STEP-1
2190 MOVE m,50+y
2200 DRAWR 8,8,1
2210 NEXT m,x
2220 TAGOFF
2230 SCREEN 1
2240 RETURN
2250 '
2260 'Vorankuendigung
2270 '
2280 MODE 0
2290 LOCATE 2,11
2300 PRINT"Ein Beispiel fuer:"
2310 LOCATE 9-LEN(cmd)/2,13
2320 PAPER 4
2330 PRINT CHR$(24)" "cmd" "CHR$(24)
2340 PAPER 0
```

```
2350 FOR m=1 TO 1500
2360 IF INKEY$=""THEN NEXT
2370 RETURN
2380 '
2390 'Bildschirm loeschen
2400 '
2410 PRINT CHR$(7);
2420 FOR m=1 TO 4000
2430 IF INKEY$=""THEN NEXT
2440 x=10+flag
2450 y=flag-25*(flag=0)
2460 flag=flag XOR 1
2470 :
2480 LOCATE 1,y
2490 FOR m=1 TO 25
2500 PRINT CHR$(x);
2510 CALL &BD19
2520 NEXT
2530 RETURN
```

SOFTWARE-TEST ✣ SOFTWARE-TEST ✣

TURBO-PASCAL 3.0 für Schneider CPC 464/664

Pascal-Programmierung auf dem Schneider ist sicher nichts Besonderes. Seit einigen Wochen ist jedoch auch die, inzwischen so beliebte, TURBO 3.0 Version für den Schneider erhältlich. Die CPU-Redaktion hat dies zum Anlaß genommen, die CPC-Version genauer unter die Lupe zu nehmen.

Was ist TURBO-PASCAL?

Turbo-Pascal ist erst seit relativ kurzer Zeit (1984) auf dem Markt erhältlich. Trotzdem hat sich diese Pascal-Version über 300 000 mal verkauft und wird inzwischen als De-facto-Standard angesehen.

Turbo-Pascal ist ein neuer Pascal-Dialekt, der einige Vorzüge gegenüber seinen Konkurrenten aufzuweisen hat. - Turbo-Pascal ist schnell und komfortabel. -

Hervorstechendes Merkmal beim Arbeiten mit Turbo-Pascal sind die kurzen Zykluszeiten bei der Programmentwicklung. Das bedeutet, daß einerseits der Compiler die Programmtexte recht schnell übersetzt und andererseits wenig Zeit benötigt wird, um bei Auftreten eines Fehlers wieder in den Programmtext zu kommen, damit der Fehler wieder korrigiert werden kann. Beim Programmieren schleichen sich häufig Fehler ein, so daß der Anwender die Schnelligkeit des Turbo-Systems schätzen lernt.

Trotzdem sollte man nicht dazu verleitet werden, sogenannte Instant-Programme ohne vorherige Planung am Bildschirm zu entwickeln. Turbo-Pascal läuft unter den Betriebssystemen MS-DOS;CP/M80 (CPM 2.2) und CP/M 86.

Der Compiler erzeugt echten Maschinencode, der zusammen mit einer 8KByte-Libery als COM-Datei abgelegt wird. Diese COM-Datei läßt sich direkt vom Diskettenbetriebssystem aktivieren, Turbo-Pascal ist hierfür nicht mehr erforderlich.

Da compilierte Programme in Maschinensprache ausgeführt werden, ist die Arbeitsgeschwindigkeit auch entsprechend schnell. Mit Turbo-Pascal steht dem Benutzer ein sehr komfortables Programmierwerkzeug zur Verfügung. Es umfaßt deutlich mehr Standard-Funktionen und Prozeduren als das Standard-Pascal und die meisten anderen Dialekte. Außerdem arbeitet es mit einer großen Rechengenauigkeit und kennt einige zusätzliche Typen und Konstanten.

Turbo-Pascal ist auf allen CP/M-Rechnern lauffähig sowie unter MS-DOS. Daher sind die Programme weitgehend ohne Veränderungen übertragbar, und „Besonderheiten" des Rechners spielen keine Rolle.

Darstellung 2: Reservierte Wörter

(Es ist unerheblich, ob die Wörter groß oder klein geschrieben werden)

ABSOLUTE	CASE	DOWNTO	FILE
AND	CONST	ELSE	FOR
ARRAY	DIV	END	FORWARD
BEGIN	DO	EXTERNAL	FUNCTION
GOTO	NOT	REPEAT	TYPE
IF	OF	SET	UNTIL
IN	OR	SHL	VAR
INLINE	PACKED	SHR	WHILE
LABEL	PROCEDURE	STRING	WITH
MOD	PROGRAMM	THEN	XOR
NIL	RECORD	To	

Standard-bezeichner

ARCTAN	EOF	LOWVIDEO	RENAME
ASSIGN	EOLN	LST	RESET
AUX	ERASE	LSTOUTPTR	REWRITE
AUXINPTER	ERRORPTR	MARK	ROUND
AUXOUTPTR	EXECUTE	MAXINT	SEEK
BLOCKREAD	EXIT	MEM	SEEKEOF
BLOCKWRITE	EXP	MEMAVAIL	SEEKEOLN
BOOLEAN	FALSE	MOVE	SIN
BUFLEN	FILEPOS	NEW	SIZEOF
BYTE	FILESIZE	NORMVIDEO	SQR
CHAIN	FILLCHAR	ODD	SQRT

Fortsetzung von Seite 49:

CHAR	FLUSH	ORD	STR
CHR	FRAC	OUTPUT	SUCC
CLOSE	GETMEN	OVRDRIVE	SWAP
CLREOL	GOTOXY	OVRPATH	TEXT
CLRSCR	HEAPPTR	PARAMCOUNT	TRM
CON	HI	PARAMSTR	TRUE
CONINPTR	IORESULT	PI	TRUNC
CONOUTPTR	INPUT	PORT	UPCASE
CONCAT	INSLINE	POS	USR
CONSTPTR	INSERT	PRED	USRINPTR
COPY	INT	PTR	USROUTPR
COS	INTEGER	RANDOM	VAL
CRTEXIT	KBD	RANDOMIZE	WRITE
CRTINIT	KEYPRESSED	READ	WRITELN
DELLINE	LENGTH	READLN	
DELAY	LN	REAL	
DELETE	LO	RELEASE	

Darstellung 3:

Erlaubte Zeichen

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
0123456789
_(Unterstreichungszeichen)
()
[] Ersatzsymbol: (..)
{ } Ersatzsymbol: (**)
”.,;:
+-†/
=<><<=>>=
:=
^
$#

Darstellung 4: Vordefinierte Typen

INTEGER
BYTE (Unterbereich 0..255)
REAL
CHAR
STRING (mit Angabe der max. Zeichenzahl)
BOOLEAN

Vordefinierte Konstanten

PI = 3.1415926536
FALSE
TRUE
MAXINT = 32767
NIL

Vordefinierte Felder

MEM
PORT

TURBO PASCAL III - VERSION II (Grafik)

Neben dem Standard-Turbo-Paket gibt es für den Schneider CPC 464 und 664 eine erweiterte Grafikversion. Diese wird mit dem Standard-Bedienungshandbuch sowie einem Zusatzheft, welches die einzelnen Grafikfunktionen näher erläutert, ausgeliefert. Läßt man sich das Direktory der Turbo-Disk auslisten so wird man über die vielen Filenamen etwas erstaunt sein, insbesondere deswegen, weil im Handbuch nur wenig bzw. garnichts darüber ausgesagt wird. Wir haben uns deshalb entschlossen einige Files auszuprobieren.

Darstellung 1:
Direktory der Turbo-Pascal Masterdiskette

TURBO	COM	:	CPCINST	COM	:	READ	ME	:	WINDOW	PAS
CMDLIN	PAS	:	TURBO	MSG	:	LISTER	PAS	:	MC	HLP
MC-MOD06	INC	:	MC-MOD07	INC	:	MC-MOD04	INC	:	MC-MOD05	INC
MCDEMO	MCS	:	TURBO	OVR	:	MC-MOD00	INC	:	MC-MOD01	INC
MC-MOD03	INC	:	MC	PAS	:	MC-MOD002	INC	:	ART	PAS
TURTLE	PAS	:	GRAFIK	INC	:	TURTLE	INC			

Bedeutung und Aufgabe der einzelnen Files:

TURBO.COM
Dies ist das eigentliche Turbo-Pascal Hauptprogramm. Es belegt etwa 32 kBytes RAM und wird von CP/M aus mit dem Kommando TURBO aufgerufen. Die einzelnen Funktionen sind im Handbuch genau dokumentiert.

CPCINST.COM
Dieses Programm ersetzt das eigentliche Installationsprogramm, da alle Funktionen schon vom Hersteller definiert wurden. Mit CPCINST lassen sich nur noch die Farbkombinationen verändern, was wohl auch nötig ist, da die eingestellte Kombination (greller Blauton) nicht gerade augenschonend ist. Auf einem Farbmonitor ist die Kombination grau/schwarz relativ brauchbar.

READ.ME
Dies ist ein Textfile, das besondere Informationen über die CPC-Turbo-Version enthält. Dieses File kamm mit dem Kommando TYPE READ.ME auf Schirm oder Drucker ausgelistet werden.

TURBO.MSG
Dieses File enthält alle Fehlermeldungen in deutschem Klartext.

LISTER.PAS
Mit diesem Programm können Pascal-Quelltexte ausgedruckt werden. Die Druckbreite sowie einige andere Parameter können im Quelltext geändert werden. Bevor das Programm genutzt werden kann, muß es jedoch compiliert werden.

MC.HLP
Dies ist ein Textfil welches Informationen zu dem Programm Mirco-Calc enthält. Dieses Programm ist als Demo für die Leistungsfähigkeit des Pascal-Systems mit auf der Diskette enthalten.

MC.PAS/MCDEMO/MC-MODO0 bis MC-MODO7
Mirco-Calc Teilprogramme. Ein hervorragendes Kalkulationsprogramm ähnlich dem bekannten Visicalc. Auch dieses Programm muß erst compiliert werden.

GRAFIK.INC
Dies ist eine spezielle Grafik-Erweiterung für den Schneider CPC 464. (Wir gehen noch näher darauf ein.)

TURTLE.INC
Dies ist die bekannte TURTLE-GRAFIK Erweiterung für den CPC. (Wir gehen noch näher darauf ein.)

TURTLE.PAS
Ein Demoprogramm zu der Turtle-Grafikerweiterung.

ART.PAS
Ein Demoprogramm zu der speziellen Grafikerweiterung.

Grafik in Turbo-Pascal

Das Grafikpaket des Schneiders-CPC besteht aus zwei wesentlichen Modulen:
Basis-Paket GRAFIC.INC und TURTLE.INC

Das Basis-Paket ermöglicht die Ansteuerung aller wesentlichen Firmware-Grafik-Routinen, wie „Ziehen einer Linie", „Bewegen des Grafikcursors", „Ändern der Farbe" u.v.m..
Leider vermißt man auch in Pascal Befehle zum Zeichnen und Bewegen von Shapes. Dies wäre sicherlich ohne große Umstände zu realisieren gewesen. Wenn man bedenkt, daß die Grafikerweiterung etwa 60 DM kostet, so hätte man diese etwas komfortabler gestalten können. Die Turtle-Grafik benutzt die Basis-Grafik und stellt eine komplette Implementierung dieses populären Grafikinterfaces dar. Leider vermißt man eine Dokumentation mit der Auflistung der einzelnen Funktionen. Offensichtlich wurde davon ausgegangen, daß der Befehlssatz von Turtle-Grafik schon weitgehend bekannt ist. Nur einige Seiten des Standard-Handbuches befassen sich mit der IBM-Turtle-Grafik, die jedoch teilweise ähnlich wie das CPC Turtle arbeitet.
Um diese Routinen in eigene Programme einfügen zu können, muß am Anfang des Programms ein Einladebefehl für ein Includefile stehen. Soll beispielsweise die GRAFIK-Erweiterung verwendet werden, so setzen wir an den Anfang des Programms einfach:
(*$I GRAFIC.INC *)

Befehlsübersicht der Grafik-Erweiterung:

CIRCLE	KREIS ZEICHNEN
CLG	GRAFIKFENSTER LÖSCHEN
CURSORON	TEXTCURSOR WIRD EINGESCHALTET
CURSOROFF	TEXTCURSORT WIRD AUSGESCHALTET
DRAW	LINIE ZIEHEN (WIE IN BASIC)
DRAWR	LINIE ZIEHEN (RELATIV)
FILLCIRCLE	GEFÜLLTEN KREIS ZEICHNEN
GETGRAFMODE	GRAFIKMODUS FESTSTELLEN
GRAFDELAY	BILDSCHIRMAUSGABE WIRD VERLANGSAMT
GRAFMODE	FARBSTIFTMODUS (NORMAL,XOR,AND,OR)
GRAVEMOVE	GRAFIKCURSOR POSITIONIEREN
GRAVEMOVE	GRAFIKCURSOR POSITIONIEREN (RELATIV)
GRAFOUT	ZEICHEN WIRD AN GRAFIKPOSITION AUSGEGEBEN
GRAFPAPER	GRAFIK HINTERGRUNDFARBE-FARBE WÄHLEN
GRAFPEN	GRAFIK STIFTFARBE WÄHLEN
INITCALL	REGISTER DES Z80 UMSETZEN
MODE	BILDSCHIRMMODUS WÄHLEN
ORIGIN	GRAFIKFENSTER DEFINIEREN
PAPER	TEXT HINTERGRUND-FARBE WÄHLEN
PEN	TEXT STIFTFARBE WÄHLEN
PLOT	PUNKT SETZEN
PLOTR	PUNKT SETZEN (RELATIV)
SPEEDKEY	TASTATUR-ABFRAGEGESCHWINDIGKEIT WIRD GEÄNDERT
TAG	UMSCHALTUNG TEXT AUF GRAFIKKOORDINATEN
TAGOFF	TAG-MODUS WIEDER ABSCHALTEN
TEST	TEST OB PUNKT VORHANDEN
TESTR	TEST OB PUNKT VORHANDEN (RELATIV)
WHEREX	SPALTENPOSITION DES TEXTCURSORS ERMITTELN
WHEREY	ZEILENPOSITION DES TEXTCURSORS ERMITTELN
XCVECTOR	GRAFIKPOSITION DURCH WINKEL ERMITTELN
YCVECTOR	GRAFIKPOSITION DURCH WINKEL ERMITTELN
XPOS	X-POSITION DES GRAFIKCURSOR ERMITTELN
YPOS	Y-POSITION DES GRAFIKCURSOR ERMITTELN
YVECTOR	WIE XCVECTOR JEDOCH RELATIV
YVECTOR	WIE YXVECTOR JEDOCH RELATIV

Nur wenig Speicherplatz für Programme

Der Schneider CPC 464/664 verfügt nur über relativ wenig Arbeitsspeicher. Für Anwenderprogramme unter CP/M stehen nur ca. 39 KByte RAM zur Verfügung. Da Turbo-Pascal selbst bereits 31 KByte Speicherplatz benötigt, lassen sich mit dem Editor nur Texte bis zu 8 KByte Größe bearbeiten. Sollen längere Programme entwickelt werden, so muß man sich mit Hilfe des Inclute-Befehls helfen. Dieser Befehl erlaubt das Zusammenfügen mehrerer Kurzprogramme zu einem Hauptprogramm. Auf diese Art ist auch das Demoprogramm Micro-Calc geschrieben worden.
Aber auch diese Art der Programmierung stößt beim Schneider oft auf die Meldung „Fehler 99", was soviel wie Speicherüberlauf bedeutet. Man kann diese Meldung verhindern, indem im Grundmenü die Option „C" für COM-File gewählt wird. Dadurch wird während des Compilierens das Objekt-File auf Diskette geschrieben. Die Compilierzeit verlängert sich dadurch natürlich entsprechend, hält sich jedoch Dank des schnellen Diskettenlaufwerkes in Grenzen.

Unsere Erfahrung mit dem System

Turbo-Pascal ist sicher eines der leistungsstärksten und vielseitigsten Pascal-Versionen auf

Fortsetzung auf S. 52

dem Markt.
Die enorme Geschwindigkeit compilierter Programme erlaubt auch die Programmierung von Action-Games, die unter CP/M laufen können. Etwas nachteilig zeigte sich der geringe Arbeitsspeicher des Schneider CPC. Die Frage, ob Turbo-Pascal auch mit einer RAM-Erweiterung lauffähig ist, werden wir in einer nächsten Ausgabe klären.

Vertrieb:
Heimsoeth/München
Weeske/Backnang
u.v.a.

Frank Brall

Achten Sie auf unsere günstigen Softwareangebote!
CPU-Software-Katalog
auf Seite 88/89.

Turbo-Pascal 3.0 * Turbo-Pascal 3.0 * Turbo-Pascal 3.0

ASCII-LIST für Epson MX80TypIII/MX82/FX80

Pascal Programme verwenden sehr oft spezielle ASCII-Zeichen wie beispielsweise die ekkigen Klammern. Da heute viele Drucker auf den deutschen Zeichensatz eingestellt sind, können diese Zeichen nicht ausgedruckt werden. Um diesen Nachteil zu beheben, gibt es zwei Möglichkeiten:
1. Drucker öffnen und umschalten. (Nachteil: keine Umlaute für Texte)
2. Das folgende Programm eingeben.

Das Programm wurde in Turbo-Pascal auf einem CPC-464 geschrieben, dürfte jedoch auch auf jedem anderen Rechner lauffähig sein.
Nach dem Start dieses Programms wird man nach dem Filenamen des zu druckenden Textes gefragt. Wichtig ist, daß dieser als ASCII-Datei auf einer Diskette abgelegt ist. Darauf kann man noch zwischen Normal- und Fettschrift wählen.
Das Programm druckt nun das angegebene File inclusive den speziellen ASCII-Zeichen auf dem Drucker und dem Bildschirm aus.
Turbo-Pascal erlaubt zur Kennzeichnung von Kommentaren auch das Benutzen der geschweiften Klammern. Da die Klammern im Epson-Zeichensatz nicht enthalten sind, werden diese in den bekannten Standard " (* " und " *) " umgewandelt.

Frank Brall

Turbo 3.0 Version on CP/M 80 for CPC 464

```
program list; (* Utility fuer Turbo-Pascal Programmierer *)
       (* Pascal-Textdatei auf Epson MxBØTypIII ausgeben *)
       (* verwendet wird der Original ASCII Zeichensatz  *)
       (* mit allen Sonderzeichen []/@ *)
       (* Der Deutsche Printer-Zeichensatz muss nicht abgeschaltet werden *)
       (* Autor Frank Brall 13.10.85 *)
  var
       name    :string[20];
       t       :text;
       zeichen :char;
       ok      :boolean;

  procedure bitBon;
    begin
      write(lst,chr(27),chr(62));
    end;

  procedure bitBoff;
    begin
      write(lst,chr(27),chr(61));
    end;

begin
  clrscr;
  writeln(' **** LIST ****      Copyright Frank Brall  1985');
  writeln;
  writeln;
  write ('Filenamen ? ');
  readln (name);
```

In der nächsten „CPU"!

- *Fortsetzung der Hardware-Serie: Universelles Bussystem für VC-20/C-64 (Teil 4)*
- *Spectrum-Kurs (Teil 2)*
- *Test: Die neue Floppygeneration von Commodore*

```
assign (t,name);
(*$i-*) reset (t); (*$i+*)
ok := ioresult=0;
if not ok then writeln('Datei ist nicht auf Diskette !');
if ok then
  begin
    writeln;
    write('<f> Fettdruck  <n> Normaldruck  ?');
    read(zeichen);
    if zeichen = 'f' then write(lst,chr(27),chr(69));
    if zeichen = 'n' then write(lst,chr(27),chr(70));
    writeln(lst,'Turbo 3.0 Version on CP/M 80 for CPC 464    (c) F.Brall');
    writeln(lst);
    writeln(lst);
    while not eof(t) do
      begin
        read(t,zeichen);
        case zeichen of
          '[':zeichen := chr(130);
          ']':zeichen := chr(132);
          '\':zeichen := chr(131);
          '@':zeichen := chr(133);
          '{':begin write(lst,'('); zeichen := '*' end;
          '}':begin write(lst,'*'); zeichen := ')' end;
        end (* case *);
        if ord(zeichen) > 128 then
          begin
            zeichen := chr(ord(zeichen)-128);
            bit8on;
          end
          else  bit8off;
        write(lst,zeichen);
        write (zeichen);
```

```
      end (* while eof *);
    end (* ok *);
  close(t);
end.
```

Epson FX/RX-80-Drucker druckt Atari Zeichensatz

Dieses Programm versetzt den Epson FX/RX-80-Drucker in die Lage, das gesamte Zeichenspektrum des Atari-Zeichensatzes auszudrucken.

Dieses Programm basiert auf der Graphikfähigkeit des EPSON-Druckers. Das Maschinenprogramm, das vollständig in PAGE 6 hineinpaßt, enthält die Druckerroutinen, die das auszugebende Zeichen aufbereiten und es an den Drucker senden. Angesprochen wird die neue Fähigkeit durch ein neudefiniertes Gerät 'A:', das im wesentlichen genauso gehandhabt wird, wie das schon vorhandene Gerät 'P:'. Der BASIC-Befehl ' OPEN #1,4,0,"A:" ' läßt sich ebenso benutzen, wie die Möglichkeit eines Ausdrucks mit ' LIST"A: '. Durch das Poken in die Speicherzelle 1741 läßt sich die Größe der Zeichen variieren. Der Wert 4 entspricht normaler (80 Zeichen/Zeile), der Wert 0 gedehnter (60 Zeichen/Zeile) und der Wert 1 schmaler (120 Zeichen/Zeile) Zeichenbreite (siehe DEMO 1). Durch das Poken in die Speicheradresse 756 kann bekanntlich auf einen Alternativzeichensatz umgeschaltet werden. Automatisch paßt sich der Drucker an diesen neuen Zeichensatz an, so daß man auf vor- oder selbstdefinierte Zeichensätze zugreifen kann, um so eine Vielzahl von Zeichen zu erhalten (siehe DEMO 2 für XL-Modelle).

Nach Eingabe und Start des Programms existiert das neue Gerät unter der Bezeichn. 'A:'. Das BASIC-Programm kann nun gelöscht werden. Bei einem RESET geht die Geräteeintragung verloren. Sie kann durch das Ablaufenlassen der Zeilen 70 bis 100 neu eingetragen werden.

```
10 .TAB 11,1,26
20 .OPT LIST
30 .OPT NO EJECT
40 .OPT OBJ
50 ;
60 ; ATARI ZEICHEN
SATZ AUF DEM RX/FX
80-DRUCKER
70 ; (C) 1.9.1985
BY MICHAEL SASSIN
80 ;
```

ASSEMBLER-LISTING

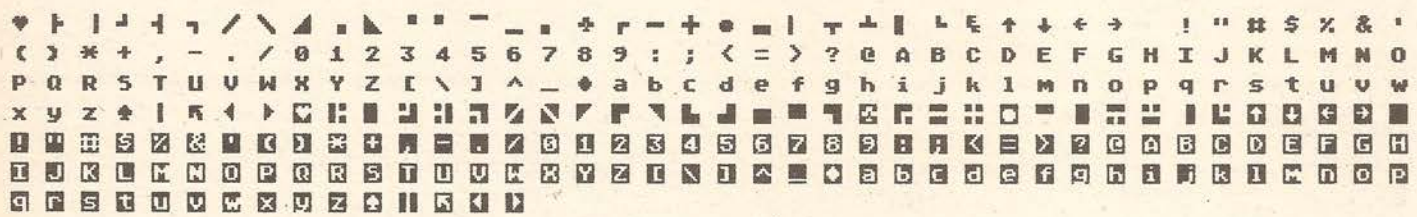
```
     =00D4          90 ZEICHENS = $D4
     =00D6          0100 MASKE  = $D6
     =00D8          0110 ZEICH  = $D8
     =0072          0120 XREG   = $72
                    0130 ;
0000                0140        *= $0600
0600 A672           0150 BEG    LDX XREG          LADE INDEX
0602 9DD206         0160        STA BUFFER,X      SPEICHERE AKKU IN DEN BUFFER AB
0605 C99B           0170        CMP #$9B          WAR DAS ZEICHEN <RETURN> ?
0607 F00A           0180        BEQ DRUCK1        SPR. WENN JA
0609 E8             0190        INX               ERHOEHE INDEX
060A 8672           0200        STX XREG          SPEICHERE INDEX AB
060C E014           0210        CPX #20           20. POSITION ERREICHT ?
060E F012           0220        BEQ DRUCK         SPR. WENN JA
0610 A001           0230 BEG1   LDY #1            CIO-KENNUNG : ERFOLGREICHE OPERATION
0612 60             0240        RTS               ZURUECK ZUM CIO
                    0250 ;
0613                0260 DRUCK1
                    0270 ;
0613 202206         0280        JSR DRUCK         BUFFER AUSDRUCKEN
0616 A200           0290        LDX #0            INDEX AUF 0 SETZEN
0618 8672           0300        STX XREG
061A A90A           0310        LDA #10           ZEILENVORSCHUB AN DRUCKER AUSGEBEN
061C 20AE06         0320        JSR PRINT
061F A001           0330        LDY #1            ZURUECK ZUM CIO
0621 60             0340        RTS
                    0350 ;
0622                0360 DRUCK
                    0370 ;
0622 8A             0380        TXA               INDEX POSITION IN AKKU
0623 0A             0390        ASL A             MULTIPLIZIERE MIT 8
0624 0A             0400        ASL A
0625 0A             0410        ASL A
0626 8DCC06         0420        STA TAB+1         SPEICHERE ERGEBNIS IN TAB+1
0629 209E06         0430        JSR INIT          INITIALISIERE DRUCKER
062C A200           0440        LDX #0            ZAEHLER AUF 0
062E 8ED106         0450        STX ZAEHLER
0631 AED106         0460 DRU1   LDX ZAEHLER       LADE ZAEHLER
0634 E472           0470        CPX XREG          VERGLEICHE MIT MAXIMALER LAENGE
0636 D007           0480        BNE DRU2          SPR. WENN BUFFERENDE NICHT ERREICHT
0638 A200           0490        LDX #0            ZURUECK ZUM CIO
063A 8672           0500        STX XREG
063C A001           0510        LDY #1
063E 60             0520        RTS
063F A900           0530 DRU2   LDA #0             LOESCHE SPEICHERZELLE ZEICHENS+1
0641 85D5           0540        STA ZEICHENS+1
0643 BDD206         0550        LDA BUFFER,X      NAECHSTES ZEICHEN AUS DEM BUFFER
0646 1007           0560        BPL DRU3          SPR. WENN AKKU<128
0648 A001           0570        LDY #1            INVERSE-FLAG SETZEN
064A 8CD006         0580        STY INV
064D 297F           0590        AND #$7F          MSB ZURUECKSETZEN
064F C960           0600 DRU3   CMP #96           DIESE ROUTINE ERMITTELT DIE
0651 B00B           0610        BCS DRU5          TATSAECHLICHE POSITION DER ZEICHEN
0653 C920           0620        CMP #32           IM ATARI-ZEICHENSATZ
0655 9005           0630        BCC DRU4
0657 E920           0640        SBC #32
0659 4C5E06         0650        JMP DRU5
065C 6940           0660 DRU4   ADC #64
```

```
065E 0A          0670 DRU5   ASL A           MULTIPLIZIERE AKKU MIT 8
065F 0A          0680        ASL A           UND SHIFTE UEBERTRAGSBIT
0660 26D5        0690        ROL ZEICHENS+1  IN DIE SPEICHERZELLE ZEICHENS+1
0662 0A          0700        ASL A
0663 26D5        0710        ROL ZEICHENS+1
0665 85D4        0720        STA ZEICHENS    SPEICHERE REST IN ZEICHENS
0667 A5D5        0730        LDA ZEICHENS+1  ERHOEHE ZEICHENS+1 UM DIE
0669 6DF402      0740        ADC 756         ZEICHENSATZ-BASIS-ADRESSE
066C 85D5        0750        STA ZEICHENS+1
066E A980        0760        LDA #$80        BIT-MASKE AUF MSB
0670 85D6        0770        STA MASKE
0672 A000        0780 DRU7   LDY #0
0674 B1D4        0790 DRU8   LDA (ZEICHENS),Y GREIFE AUF 1-BIT INFORMATION
0676 25D6        0800        AND MASKE       DURCH MASKIERUNG ZU
0678 C901        0810        CMP #1          VERGLEICHE MIT 1
067A 26D8        0820        ROL ZEICH       VERGLEICHSERGEBNIS ABSPEICHERN
067C C8          0830        INY
067D C008        0840        CPY #8          ENDE ERREICHT ?
067F D0F3        0850        BNE DRU8        SPR. WENN NEIN
0681 A5D8        0860        LDA ZEICH       LADE INFORMATION
0683 ACD006      0870        LDY INV         INVERTIERTES ZEICHEN ?
0686 C000        0880        CPY #0
0688 F002        0890        BEQ DRU9        SPR. WENN NEIN
068A 49FF        0900        EOR #$FF        INVERTIERE INFORMATION
068C 20AE06      0910 DRU9   JSR PRINT       DRUCKE INFORMATION
068F 46D6        0920        LSR MASKE       UNTERSUCHE NAECHSTES BIT
0691 90DF        0930        BCC DRU7        SPR. WENN NOCH NICHT 8 BIT UNTERSUCHT
0693 EED106      0940        INC ZAEHLER     ERHOEHE ZAEHLER
0696 A900        0950        LDA #0          LOESCHE INVERSE-FLAG
0698 8DD006      0960        STA INV
069B 4C3106      0970        JMP DRU1        NAECHSTES ZEICHEN DRUCKEN
                 0980 ;
069E             0990 INIT
                 1000 ;
069E A205        1010        LDX #5          ANZAHL DER STEUERZEICHEN
06A0 8A          1020 INI1   TXA             RETTE X-REGISTER AUF STACK
06A1 48          1030        PHA
06A2 BDCA06      1040        LDA TAB-1,X     LADE INFORMATION
06A5 20AE06      1050        JSR PRINT       DRUCKE SIE
06A8 68          1060        PLA             HOLE X-REGISTER VOM STACK
06A9 AA          1070        TAX
06AA CA          1080        DEX             ERNIEDRIGE X-REGISTER
06AB D0F3        1090        BNE INI1        SPR. WENN NICHT 0
06AD 60          1100        RTS
                 1110 ;
06AE             1120 PRINT
                 1130 ;
06AE A200        1140        LDX #0
06B0 A8          1150        TAY             RETTE AKKU IN Y-REGISTER
06B1 AD37E4      1160        LDA $E437       LADE ADRESSEN FUER DIE
06B4 48          1170        PHA             1-BYTE AUSGABE DES DRUCKERS AUF
06B5 AD36E4      1180        LDA $E436       DEN STACK UND SPRINGE SIE AN
06B8 48          1190        PHA
06B9 98          1200        TYA             Y-REGISTER ZURUECK IN AKKU
06BA 60          1210        RTS
                 1220 ;
06BB             1230 OPEN
                 1240 ;
```

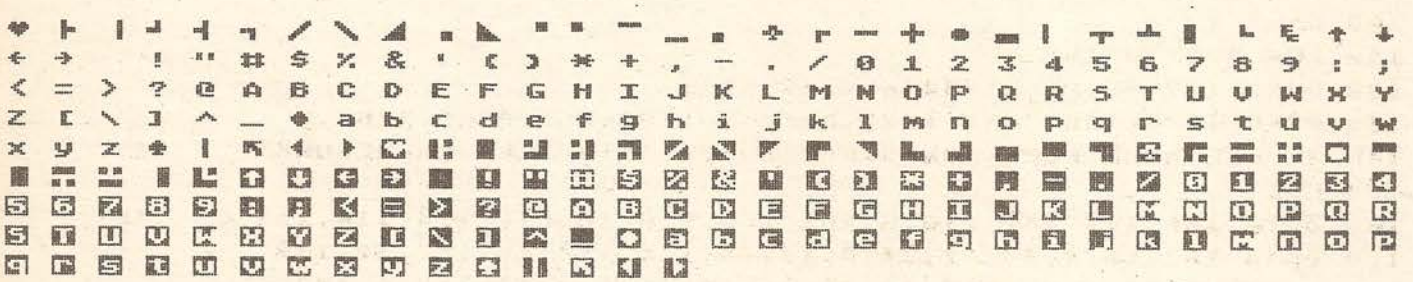
```
06BB A200       1250        LDX #0          LOESCHE INDEX UND INVERSE-FLAG
06BD 8672       1260        STX XREG
06BF 8ED006     1270        STX INV
06C2 AD31E4     1280        LDA $E431       LADE ADRESSE FUER DEN OPEN-BEFEHL
06C5 48         1290        PHA             AUF DEN STACK UND SPRINGE SIE AN
06C6 AD30E4     1300        LDA $E430
06C9 48         1310        PHA
06CA 60         1320        RTS
                1330 ;
                1340 ;
06CB 0000042A   1350 TAB    .BYTE 0,0,4,42,27 DRUCKERINITIATIONSDATEN
06CF 1B
                1360 ;
06D0            1370 INV
06D0            1380        *= *+1
06D1            1390 ZAEHLER
06D1            1400        *= *+1
06D2            1410 BUFFER
```

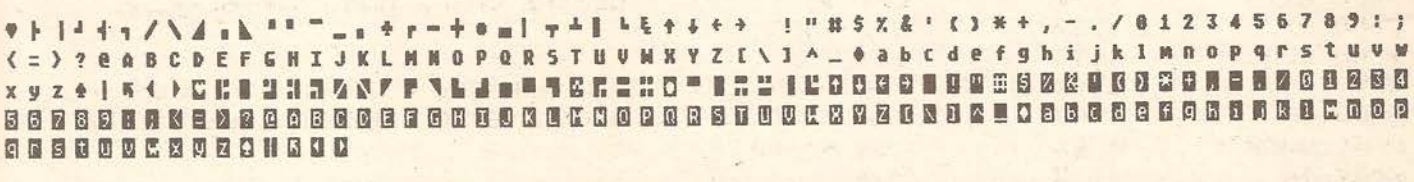
DEMOPROGRAMM

```
10 OPEN #1,4,0,"A:"
20 GOSUB 40:POKE 1741,0:GOSUB 40:POKE 1741,1:GOSUB 40:POKE 1741,4
30 END
40 ? #1:FOR X=0 TO 255:IF X=155 THEN X=X+1
50 ? #1;" ";CHR$(X);:NEXT X:? #1;CHR$(155):RETURN
```



DEMOPROGRAMM





```
DEMOPROGRAMM 2

10 OPEN #1,4,0,"A:"
20 POKE 756,204
30 FOR X=0 TO 255:IF X=155 THEN X=X+1
40 ? #1;" ";CHR$(X);:NEXT X
```


BASIC-LOADER

```
10 REM EPSON FX/RX 80-DRUCKER DRUCKT ATARI-ZEICHENSATZ
20 REM (C) 9.1985 BY MICHAEL SASSIN
30 REM
40 B=0
50 FOR X=1536 TO 1743:READ A:POKE X,A:B=B+A:NEXT X
60 IF B<>23404 THEN ? "DATAFEHLER":END
70 FOR Y=794 TO 831 STEP 3
80 IF PEEK(Y)=ASC("A") THEN GOTO 110
90 IF PEEK(Y)=0 THEN POKE Y,ASC("A"):POKE Y+1,240:POKE Y+2,6:GOTO 110
100 NEXT Y
110 FOR Y=0 TO 15
120 POKE 1776+Y,PEEK(58416+Y):NEXT Y
130 POKE 1776,186:POKE 1777,6:POKE 1782,255:POKE 1783,5
140 ? "DIE NEUE FUNKTION STEHT UNTER 'A:' ZUR VERFUEGUNG"
150 END
160 DATA 166,114,157,210,6,201,155,240,10,232,134,114,224,20,240
170 DATA 18,160,1,96,32,34,6,162,0,134,114,169,10,32,174
180 DATA 6,160,1,96,138,10,10,10,141,204,6,32,158,6,162
190 DATA 0,142,209,6,174,209,6,228,114,208,7,162,0,134,114
200 DATA 160,1,96,169,0,133,213,189,210,6,16,7,160,1,140
210 DATA 208,6,41,127,201,96,176,11,201,32,144,5,233,32,76
```

```
220 DATA 94,6,105,64,10,10,38,213,10,38,213,133,212,165,213
230 DATA 109,244,2,133,213,169,128,133,214,160,0,177,212,37,214
240 DATA 201,1,38,216,200,192,8,208,243,165,216,172,208,6,192
250 DATA 0,240,2,73,255,32,174,6,70,214,144,223,238,209,6
260 DATA 169,0,141,208,6,76,49,6,162,5,138,72,189,202,6
270 DATA 32,174,6,104,170,202,208,243,96,162,0,168,173,55,228
280 DATA 72,173,54,228,72,152,96,162,0,134,114,142,208,6,173
290 DATA 49,228,72,173,48,228,72,96,0,0,4,42,27
```

LESERUMFRAGE ✤ LESERUMFRAGE ✤ LESERUMFRAGE

Liebe Leser!

Der Computer-Markt ist in den letzten Monaten sehr stark in Bewegung geraten. Immer neue Computermodelle und Computer-Zeitschriften drängen in den Handel. Enorme Preisstürze lassen schon lange tot geglaubte Computersysteme wieder aufleben.
Aus diesem Grund hat sich die CPU entschieden eine Leserumfrage durchzuführen. Nach der Auswertung dieser Umfrage hoffen wir unsere Zeitschrift für den Leser noch attraktiver zu gestalten.

Ihre Mühe belohnen wir durch eine **kostenlose** Kleinanzeige in unserer Zeitschrift CPU. Zusätzlich erhält jeder 10. Einsender eine **Gratis-Kassette** für das gewünschte System.

Mit freundlichem Gruß
Ihre CPU Redaktion

Bitte senden Sie den ausgefüllten Fragebogen an:
Tronic Verlag /
CPU-Redaktion
Kennwort „Umfrage“
Landstraße 29
3444 Wehretal

CPU-Fragebogen

Name: _____ Alter: ____ Jahre

Straße: _____ PLZ/ORT: _____

Ich interessiere mich für Computer seit (Datum): _____

Ich besitze folgendes Computersystem(e): _____

Ich plane den Kauf des Computersystems: _____

Ich interessiere mich für:
- Utilitys und Tips: ☐
- Anwenderprogramme: ☐
- Spiele: ☐
- Programm-Vorstellungen: ☐
- Hardware allgemein: ☐
- Aktuelle Informationen: ☐
- Sonstiges: _____

Ich lese folgende Computerzeitschriften regelmäßig:

_____

_____

Ich lese folgende Computerzeitschriften gelegentlich:

_____

_____

Anzahl der selbst abgetippten Listings pro Monat:
- keine ☐
- gelegentlich ☐
- 1 bis 2 Listings ☐

3 bis 5 Listings ☐ mehr als 5 Listings ☐

In dieser Ausgabe der CPU gefiel mir besonders:

Utility: _____

User: _____

Tips: _____

Sonstiges: _____

Ich wünsche mir mehr Informationen/Listings für folgende Themen:

_____

_____

*** **GRATIS!** ****

Sollte ich der 10. Einsender sein, so senden Sie mir bitte die Kassette mit der Bestellnummer: _____ zu.

Bitte veröffentlichen Sie die nachfolgende Kleinanzeige in Ihrer nächstmöglichen Ausgabe! (Coupon auf Seite ...!)

Dieses Programm ermöglicht es, kurze Maschinenspracheroutinen in DATA-Statements umzuwaneln und in BASIC-Listings einzuflechten.

Atari: Translater für MS-Routinen

System Requirement
- ATARI 400/800 // 600XL/ 800XL Computer
- ATARI 410 // 1010 Program Recorder
- Option ATARI 810 // 1050 Disk Drive
- Option Printer

Bedienungsanleitung
1. Maschinencode laden (L Option im ATARI DUP Vers. 2.OS. String)
2. Ins BASIC gehen
3. Programm mit RUN"D:-TRANSLAT.BAS" starten
4. Den umzuwandelnden Speicherbereich eingeben (Unter, obere Grenze)
5. Zeilennummer eingeben, in denen die DATA-Statements stehen sollen.
6. Ausgabegerät eingeben. Bei RETURN ohne Eingabe Ausgabe nur am Screen

Bei der Eingabe eines Filenamens werden die DATA-Zeilen gelistet. Sie können dann wieder mit ENTER geladen werden.

Beispielroutine für Umwandlung DATA-Speicher:

10 RESTORE 100:I=I+1:REM ZEILENNUMMER DES ERSTEN DATA-STATEMENT I=
STARTADDRESSE
20 READ A:IF A=-1 THEN 40
30 POKE I,A:GO TO 20
40 REM WEITER IM PROGRAMM

Anmerkung:
Bei Benutzung des obrigen Programms muß in das letzte DATA-Statement als letzte Zahl eine -1 hinzugefügt werden.

```
 11 REM by Klaus Rupprecht
 12 REM (c)   1985
 20 REM
 30 REM
 39 DIM P$(14):ST=0
 40 ? "{CLR}   TRANSLATER "
 41 ? "{DOWN}{DOWN}BEREICH EINGEBEN"
 42 ? "{DOWN}{DOWN}von Speicherzelle :";:INPUT A
 43 ? "{DOWN}{DOWN}bis Speicherzelle :";:INPUT B
 44 ? "{DOWN}{DOWN}erste Zeilennummer:";:INPUT ZEI
 45 ? "{DOWN}{DOWN}Ausgabegeraet     :";:INPUT P$
 46 IF P$="" THEN ST=1
 47 IF ST=1 THEN 59
 50 OPEN #1,8,0,P$
 59 T=1:DIM A$(50):GOSUB 110
 60 FOR I=A TO B
 75 P=PEEK(I)
 80 A$(LEN(A$)+1)=STR$(P):T=T+1
 81 IF T=11 THEN GOSUB 100:GOTO 90
 82 A$(LEN(A$)+1)=","
 90 NEXT I
 95 A$(LEN(A$)+1)="0":GOSUB 100
 99 END
100 IF ST=1 THEN 102
101 ? #1,A$:? A$:GOTO 110
102 ? A$
110 A$=STR$(ZEI):A$(LEN(A$)+1)=" DATA ":
120 RETURN                          T=1:ZEI=ZEI+1
```

Dies ist das erste Atari-520 ST-Programm, das wir in der „CPU" veröffentlichen!

Ableitungen für den ATARI 520 ST

von Klaus Schneider und Oliver Steinmeier

Für den neuen Atari 520 ST gibt es bisher noch nicht viel Software, aus diesem Grund bringen wir hier ein kurzes Anwenderprogramm, daß in der Mathematik sehr häufig eingesetzt werden kann.
Es handelt sich um ein Programm zur Bestimmung von beliebig vielen Ableitungen von ganzrationalen und gebrochen rationalen Funktionen. Besonders die letztgenannte Art bereitet den Mathematikern und vor allem den Schülern viel Mühe, weil das Verfahren sehr rechenaufwendig ist. Da man in der Regel (z. B. zur Kurvendiskussion) mehrere Ableitungen einer Funktion benötigt, treten häufig kleine Fehler auf, die dann die ganze restliche Arbeit zunichte machen. Doch für den Atari ist es natürlich kein Problem, auch komplizierte Funktionen beliebig oft abzuleiten.

Das mathematische Verfahren:

Die Funktion $y = \frac{x^2 + 3x}{7x^3 + 6}$

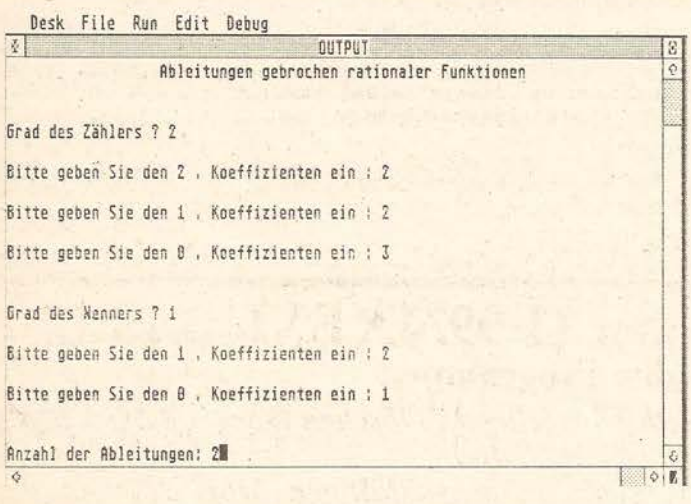
soll abgeleitet werden. Dabei geht das Programm nach der allgemein bekannten Formel

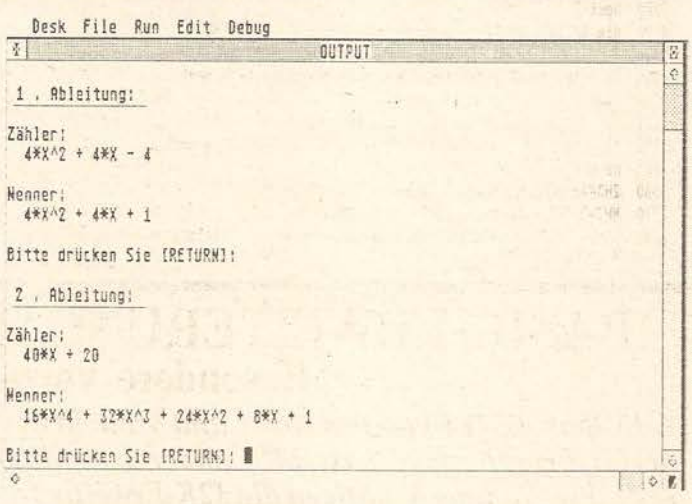
$$f'(x) = \frac{u'v - v'u}{v^2}$$

vor.

Der Anwender geht nach dem Start des Basicprogramms so vor: Zunächst erfolgt die Eingabe des Zählergrades. In dem oben aufgeführten Beispiel wäre „2" einzugeben. Dann folgt die Eingabe der Koeffizienten. Hier wäre dann „1", „3" und „0" einzugeben.
Entsprechend werden die Werte des Nenners der Funktion eingegeben.
Handelt es sich um eine ganzrationale Funktion (z. B. $f(x) = 3x^2 + 6x + 7$), so muß als Nennergrad „0" und als 0.Koeffizient „1" eingegeben werden.
Anschließend wird die Zahl der gewünschten Ableitungen festgelegt. Dann folgt die Berechnung und Ausgabe der Ableitungen. Nach der Ausgabe einer Ableitung wartet das Programm auf das Drücken der „RETURN"-Taste. Währenddessen hat man Zeit, das Ergebnis zu notieren.
Die Ableitungen sind übrigens so weit als möglich durchgekürzt und haben so die einfachste mögliche Form.
Vor dem Programmstart muß man den GRAPHICS BUFFER ausschalten, um zusätzlichen Speicherplatz zu erhalten.

❀ ❀ ❀ ❀






```
10   '*****************************************************************
20   '*                                                               *
30   '*    Ableitungen gebrochen rationaler Funktionen für ATARI 520 ST *
40   '*                                                               *
50   '*          Copyright 1985 by O. Steinmeier und K. Schneider     *
60   '*                                                               *
70   '*****************************************************************
80   '
90   '************************** Hauptprogramm **************************
100  '
110  for I=1 to 25 : 'Bildschirm löschen
120  print
130  next I
140  print tab(19)"Ableitungen gebrochen rationaler Funktionen"
150  print
160  gosub 1000 : 'Eingabe
170  gosub 2000 : 'Bestimmen der Ableitungen
180  print : input "Wollen Sie noch eine Funktion ableiten (J/N) ? ",IP$
190  if IP$="n" or IP$="N" then end else clear : goto 160
200  end
1000 '
1010 '************************ Funktion eingeben ************************
1020 '
1030 print
1040 input "Grad des Zählers ? ",ZP
1050 print
1060 dim ZP(ZP+1) : 'Dimensionierung der Zählerpolynoms auf Zählergrad + 1
1070 for I=ZP to 0 step -1 : 'Eingabe der Zählerkoeffizienten
1080 print "Bitte geben Sie den";I;". Koeffizienten ein : ";
1090 input "",ZP(I) : print
1100 next I
1110 print
1120 input "Grad des Nenners ? ",NP
1130 print
1140 dim NP(NP+1) : 'Dimensionierung der Nennerpolynoms auf Nennergrad + 1
1150 for I=NP to 0 step -1 : 'Eingabe der Nennerkoeffizienten
1160 print "Bitte geben Sie den";I;". Koeffizienten ein : ";
1170 input "",NP(I) : print
```

```
1180 next I
1190 print
1200 input "Anzahl der Ableitungen: ",AZ
1210 print
1220 return
2000 '
2010 '################## Parameter an Ableitungen übergeben ##################
2020 '
2030 ZH1=ZP : 'Grad des Zählerpolynoms für Ableitung
2040 NH1=NP : 'Grad des Nennerpolynoms für Ableitung
2050 dim ZH1(ZH1) : 'Zählerpolynom für Ableitung
2060 dim NH1(NH1) : 'Nennerpolynom für Ableitung
2070 for I=0 to ZP : ZH1(I)=ZP(I) : next I : 'Zählerpolynom übertragen
2080 for I=0 to NP : NH1(I)=NP(I) : next I : 'Nennerpolynom übertragen
2090 for K=1 to AZ
2100 print : print K;". Ableitung:" : print : linef 5,310,130,310
2110 gosub 2190 : 'Ableitung der Funktion
2120 erase ZH1,NH1
2130 dim ZH1(ZH3) : for I=0 to ZH3 : ZH1(I)=ZH3(I) : next I
2140 dim NH1(NH3) : for I=0 to NH3 : NH1(I)=NH3(I) : next I
2150 ZH1=ZH3 : NH1=NH3 : erase ZH2,ZH3,NH2,NH3 : 'alte Arrays löschen
2160 next K
2170 return
2180 '
2190 '##################### Berechnen einer Ableitung #####################
2200 '
2210 if ZH1>0 then dim ZH2(ZH1-1) : 'Ableitung des Zählers
2220 if NH1>0 then dim NH2(NH1-1) : 'Ableitung des Nenners
2230 for I=1 to ZH1 : ZH2(I-1)=ZH1(I)*I : next I : 'Zähler ableiten
2240 for I=1 to NH1 : NH2(I-1)=NH1(I)*I : next I : 'Nenner ableiten
2250 dim NH3(2*NH1)
2260 for I=0 to NH1 : 'Ableitung des Nenners: V^2
2270 for L=0 to NH1 : NH3(I+L)=NH3(I+L)+NH1(I)*NH1(L) : next L
2280 next I
2290 dim ZH3(NH1+ZH1) : 'Ableitung des Zählers: U'*V-V'*U
2300 for I=0 to ZH1-1 : 'Ableitung des Zählers: U'*V
2310 for L=0 to NH1 : ZH3(I+L)=ZH3(I+L)+ZH2(I)*NH1(L) : next L
2320 next I
2330 for I=0 to ZH1 : 'Ableitung des Zählers: V'*U
2340 for L=0 to NH1-1 : ZH3(I+L)=ZH3(I+L)-ZH1(I)*NH2(L) : next L
2350 next I
2360 ZH3=ZH1+NH1 : 'Grad der Ableitung (Zähler)
2370 NH3=2*NH1 : 'Grad der Ableitung (Nenner)
2380 while ZH3(ZH3)=0 and ZH3>0 : ZH3=ZH3-1 : wend : 'Zählergrad kürzen
2390 while NH3(NH3)=0 and NH3>0 : NH3=NH3-1 : wend : 'Nennergrad kürzen
2400 while ZH3(0)=0 and NH3(0)=0 and ZH3*NH3>0 : 'Bruchgrad kürzen
2410 for I=0 to ZH3-1 : ZH3(I)=ZH3(I+1) : next I : ZH3=ZH3-1
2420 for I=0 to NH3-1 : NH3(I)=NH3(I+1) : next I : NH3=NH3-1
2430 wend
2440 H1=ZH3(0) : H3=1 : GGT=2 : 'GGT des Zählers berechnen
2450 while abs(GGT)>1 and H3<=ZH3 : H2=ZH3(H3)
2460 while H2<>0 : GGT=H2 : H2=(H1/H2-int(H1/H2))*H2 : H1=GGT : wend
2470 H3=H3+1 : wend
2480 H3=0
2490 while abs(GGT)>1 and H3<=NH3 : H2=NH3(H3) : 'GGT Zähler & Nenner
2500 while H2<>0 : GGT=H2 : H2=(H1/H2-int(H1/H2))*H2 : H1=GGT : wend
2510 H3=H3+1 : wend
2520 if GGT=1 or GGT<>int(GGT) then 2550 : 'Falls GGT <> 1, numerisch kürzen
2530 for I=0 to ZH3 : ZH3(I)=ZH3(I)/GGT : NEXT I
2540 for I=0 to NH3 : NH3(I)=NH3(I)/GGT : NEXT I
2550 '
2560 '##################### Ausgabe der Ableitung #####################
2570 '
2580 print "Zähler:" : FLAG=0
2590 for I=ZH3 to 0 step-1
2600 if ZH3(I)=0 then 2680
2610 Z1$=right$(str$(I),len(str$(I))-1)
2620 if ZH3(I)>=0 then Z$="+" else Z$="-"
2630 Z$=Z$+str$(abs(ZH3(I)))+"*X^"+Z1$
2640 if I=ZH3 and left$(Z$,1)="+" then Z1$=right$(Z$,len(Z$)-1) : Z$=Z1$
2650 if I=1 then Z1$=left$(Z$,len(Z$)-2) : Z$=Z1$
2660 if I=0 then Z1$=left$(Z$,len(Z$)-4) : Z$=Z1$
2670 print " "Z$; : FLAG=1
2680 next I : if FLAG=0 then print " 0"
2690 print : print : print "Nenner: "
2700 for I=NH3 to 0 step-1
2710 if NH3(I)=0 then 2790
2720 N1$=right$(str$(I),len(str$(I))-1)
2730 if NH3(I)>=0 then N$="+" else N$="-"
2740 N$=N$+str$(abs(NH3(I)))+"*X^"+N1$
2750 if I=NH3 and left$(N$,1)="+" then N1$=right$(N$,len(N$)-1) : N$=N1$
2760 if I=1 then N1$=left$(N$,len(N$)-2) : N$=N1$
2770 if I=0 then N1$=left$(N$,len(N$)-4) : N$=N1$
2780 print " ";N$;
2790 next I : print : print
2800 input "Bitte drücken Sie [RETURN]: ",IP$
2810 return
```

BASIC-ERWEITERUNG für den TI-99/4A EXT. BASIC

Besondere Vorzüge des Programms:

1. Es läuft in TI Extended Basic und braucht keinerlei Erweiterung. Fast alle anderen Utility/User-Programme benötigen die 32K-Erweiterung und/oder ein Diskettenlaufwerk.

2. Es erleichtert das Programmieren auf dem TI-99 sehr. 13 neue Befehle stehen unmittelbar zur Verfügung. Sie eignen sich sowohl für Spiele-Programmierer (z. B. Boing/Crash) als auch für Profis (z. B. Mouse/Change/etc.).

3. Das Anpassen von fremden Programmen an den TI-99 wird erheblich erleichtert. In anderen Basic-Versionen übliche Befehle (z. B. Beep/Get/Left/Right/etc.) werden auf dem TI-99 simuliert.

4. Natürlich gibt es auch einiges, was sich schon viele TI-User gewünscht haben, z. B. eine Abfrage ob der Speech Synthesizer angeschlossen ist und eine Möglichkeit, das Programm vom Programm selbst ausgehend zu löschen (Verwendung z. B. als Listschutz-Falle).

5. Trotz dieser vielfältigen Möglichkeiten der BASIC-ERWEITERUNG bleiben für das eigene Programm noch rund 12 KB frei.

Das Programm BASIC-ERWEITERUNG stellt somit eine große Hilfe für alle TI-User dar.

Die 13 Befehle werden mit „CALL Name" aufgerufen und wie die normalen, eingebauten CALL-Befehle behandelt. Im einzelnen können Sie folgende Befehle verwenden:

1. ***CALL BEEP*** erzeugt einen „Piep-Ton" wie bei Input oder Accept
2. ***CALL CL*** löscht den Bildschirm, aber nicht wie Call Clear sofort, sondern langsam und „echt computermäßig" Zeichen für Zeichen einzeln
3. **CALL CRASH** erzeugt einen „Farb-Blitz" und ein typisches Explosionsrauschen (Verwendung vor allem oft in Spielen)
4. **CALL BOING(A)** erzeugt ein Gong-ähnliches Geräusch mit

der Grundfrequenz A
5. **CALL SPEECH(A)** wenn der Sprachsynthesizer angeschlossen ist, wird A gleich eins gesetzt, wenn nicht, wird A gleich null gesetzt
6. **CALL RAN(UG,OG,A)** der Variablen A wird eine Zufallszahl zwischen der Untergrenze UG und der Obergrenze OG zugeordnet
7. **CALL NEW** ACHTUNG!! Dieser Befehl simuliert NEW innerhalb eines Programms. Nach der Durchführung von CALL NEW wird das gesamte Programm unwiederbringlich gelöscht. Der Speicher ist wieder frei.
8. **CALL CHANGE(A$,B$)** Dieser Befehl ordnet der Variablen A$ den String B$ zu und der Variablen B$ den String A$ (Tausch, Verwendung vor allem in Sortier-Programmen)
9. **CALL WAIT(A)** Hält das Programm an und fährt erst fort, sobald die Taste mit dem ASCII-Code A gedrückt wurde. Gibt man für A null ein, so reicht der Druck auf eine beliebige Taste.

```
30000 !======================          BASIC ERWEITERUNG          ==================
====        TI99-4A EXT.BASIC          ======================
30010 !                    (C)1985 PATRICK SCHMITZ                AM ZEHNTHOF 4
        5480 REM.-UNKELBACH
30020 !
30030 SUB BEEP :: CALL SOUND(-111,1400,2):: SUBEND :: SUB WAIT(A)
30040 CALL KEY(0,K,S):: IF S<>1 THEN 30040 ELSE IF K=A OR A=0 THEN SUBEXIT ELSE
30040
30050 SUBEND :: SUB GET(A$)
30060 CALL KEY(0,K,S):: IF S<>1 THEN 30060 ELSE A$=CHR$(K):: SUBEXIT ELSE 30060
30070 SUBEND :: SUB SPEECH(A):: CALL PEEK(-28672,A):: IF A=96 THEN A=1 ELSE A=0
30080 SUBEND :: SUB RAN(X1,X2,A):: RANDOMIZE :: A=INT(RND*(X2-X1+1))+X1 :: SUBEN
D :: SUB NEW :: ON ERROR 30090 :: RUN ""
30090 END :: SUBEND :: SUB CHANGE(A$,B$):: X$=A$ :: A$=B$ :: B$=X$ :: SUBEND ::
SUB LEFT(A$,B$,A):: B$="" :: FOR Z=1 TO A
30100 B$=B$&SEG$(A$,Z,1):: IF Z=LEN(A$)THEN 30120
30110 NEXT Z :: SUBEXIT
30120 SUBEND :: SUB RIGHT(A$,B$,A):: B$="" :: FOR Z=1 TO A
30130 IF LEN(A$)<Z THEN 30140 ELSE B$=SEG$(A$,LEN(A$)+1-Z,1)&B$
30140 NEXT Z :: SUBEND :: SUB BOING(A):: IF A<110 THEN A=110 ELSE IF A>44733 THE
N A=44733
30150 FOR Z=1 TO 14 STEP 3 :: CALL SOUND(-100,A,Z,A+2,Z+8,A+4,Z+12):: NEXT Z ::
SUBEND :: SUB CRASH :: CALL SCREEN(16)
30160 CALL SCREEN(12):: CALL SCREEN(7):: CALL SCREEN(8):: FOR Z1=1 TO 6 STEP 2 :
: FOR Z=-5 TO -7 STEP -1 :: CALL SOUND(-200,Z,Z1):: NEXT Z :: NEXT Z1 :: SUBEND
30170 SUB CL :: FOR Z1=1 TO 24 :: FOR Z=1 TO 32 :: CALL HCHAR(Z1,Z,30):: CALL HC
HAR(Z1,Z,32):: NEXT Z :: NEXT Z1 :: SUBEND
30180 SUB MOUSE(A$):: CALL SPRITE(#1,43,16,1,17):: ZE=1 :: SP=3
30190 CALL KEY(1,K,S):: IF K<>18 THEN 30200 ELSE CALL BEEP :: CALL GCHAR(ZE,SP,A
):: A$=CHR$(A):: SUBEXIT
30200 CALL JOYST(1,X,Y):: X=SGN(X):: Y=-SGN(Y):: IF ZE+Y<1 OR ZE+Y>24 OR SP+X<1
OR SP+X>32 THEN 30190 ELSE ZE=ZE+Y :: SP=SP+X
30210 CALL LOCATE(#1,ZE*8-7,SP*8-7):: GOTO 30190 :: SUBEND
```

Change, Typ V.1.4 für Commodore C-64, C-16/116 und VC-20

Ist Ihnen dies auch schon einmal passiert: Sie haben gerade Ihre Hauptdiskette formatiert, glauben dann diesen Fehler mit einen RE-Formatiert-Programm abgestellt zu haben, und stellen dann mit Erschrecken fest, daß Ihre (mit viel Mühe erstellte) Sequentielle Datei zu einem unbrauchbaren 'PRG' FILE geworden ist.
Rettungsmethoden wie Benutzung eines Dismonitors und Änderung des FILETYP-Kennzeichens sind nur den Eingeweihten vorbehalten. Also benutzt der schlaue User das Programm 'Change Typ', spart Ärger mit verpfuschten Disketten oder langwieriges Umschreiben oder Retten des Programms. Die Benutzung ist denkbar einfach, Namen des FILES eintippen, der Rest wird blitzschnell erledigt.
Der Eintrag im Direktory wird von $82 (=PRG) in $81 (SEQ) geändert. Das reicht, um das FILE nun als leicht einzulesenden Datenblock eines Sequentiellen FILES zu kennzeichnen. Das Programm ist auch auf dem C 16/116 sowie auf dem VC-20 lauffähig. Beim C 16 müßte Zeile 90 folgendermaßen lauten:
90 COLOR 0,1 : COLOR 4,1
und beim VC 20
90 POKE 36879,0

```
C64 LISTING MIT CHECKSUMMEN (C V1.0)
10 REM CHANGE TYP V. 1.4                                    <214>
20 :                                                        <78>
30 REM       JUERGEN   OTTO                                 <74>
40 REM       4460  NORDHORN                                 <119>
50 REM       SCHUBERTSTR. 3                                 <190>
55 REM       T. 05921/15151                                 <128>
60 REM       (C)  AUGUST 85                                 <123>
70 :                                                        <128>
80 :                                                        <138>
90 POKE 53280,0 : POKE 53281,0                              <172>
100 PRINT "{CLEAR GREEN SPACE14}CHANGE{S
PACE}TYP"                                                   <178>
120 PRINT "{DOWN SPACE6 RED}(C){SPACE GR
EEN}08.85{SPACE}BY{SPACE BLUE}JUERGEN{SP
ACE}OTTO{GREEN}"                                            <58>
```

```
130 PRINT "{DOWN2}ES{SPACE}IST{SPACE}NIC
HT{SPACE}MOEGLICH,{SPACE}EIN{SPACE}'PRG'
"                                        <60>
140 PRINT "FILE{SPACE}IN{SPACE}EIN{SPACE
}'REL'{SPACE}FILE{SPACE}ZU{SPACE}AENDERN
."                                       <184>
150 PRINT "NUR{SPACE}'SEQ'{SPACE}FILES{S
PACE}KOENNEN{SPACE}ERZEUGT{SPACE}WERDEN{
DOWN}"                                   <78>
155 :                                    <213>
160 GOSUB 590 : REM NAME EINLESEN        <220>
165 :                                    <223>
170 GOSUB 320 : REM PRG SUCHEN           <183>
175 :                                    <233>
180 IF GF=1 THEN 210                     <220>
200 CLOSE2 : CLOSE1 : END                <203>
205 :                                    <7>
206 REM SCHREIBEN DES SEQ-ZEICHENS       <33>
207 :                                    <9>
210 PRINT#15,"B-P";2;X*32+2              <65>
220 GET#2,X$ : IF X$="" THEN X$=CHR$(0)  <110>
230 F=ASC(X$)                            <49>
240 F=F-1                                <13>
250 PRINT#15,"B-P";2;X*32+2              <105>
260 PRINT#2,CHR$(F);                     <4>
270 PRINT#15,"B-P";2;0                   <35>
280 PRINT#15,"U2";2;0;TT;SS              <212>
290 CLOSE2 : CLOSE15                     <136>
300 PRINT "O.K."                         <40>
310 END                                  <183>
315 :                                    <118>
316 REM SUCHEN DES NAMENS                <71>
317 :                                    <120>
320 OPEN 15,8,15,"IO":OPEN2,8,2,"#"      <47>
330 T=18 : S=1 : TT=18 : SS=1            <3>
340 PRINT#15,"B-R";2;0;T;S               <210>
350 TT=T : SS=S                          <149>
360 PRINT#15,"B-P";2;0                   <125>
370 GET#2,X$ : IF X$="" THEN X$=CHR$(0)  <5>
380 T=ASC(X$)                            <214>
390 GET#2,X$ : IF X$="" THEN X$=CHR$(0)  <25>
400 S=ASC(X$)                            <233>
410 FOR X=0 TO 7                         <32>
420 PRINT#15,"B-P";2;X*32+2              <20>
430 GET#2,X$ : IF X$="" THEN X$=CHR$(0)  <65>
440 IF ASC(X$)=0 THEN 540                <29>
450 PRINT#15,"B-P";2;X*32+5              <104>
460 FF$=""                               <73>
470 FOR Y=0 TO 15                        <173>
480 GET#2,X$ : IF X$="" THEN X$=CHR$(0)  <115>
490 IF ASC(X$)=160 THEN 510              <118>
500 FF$=FF$+X$                           <201>
510 NEXT Y                               <51>
520 IF FL THEN PRINT FF$ : GOTO 540      <215>
530 IF F$=FF$ THEN GF=1 : GOTO 580       <253>
540 NEXT X                               <80>
550 IF T<>0 THEN 340                     <199>
560 CLOSE2 : CLOSE15                     <151>
570 IF FL=0 THEN PRINT "{DOWN}FILENAME{S
PACE}NICHT{SPACE}GEFUNDEN!"              <95>
580 RETURN                               <212>
585 :                                    <133>
586 REM EINGABE DES NAMENS               <151>
587 :                                    <135>
590 INPUT "NAME{SPACE}DES{SPACE}FILES:{S
PACE}";F$                                <163>
600 GF=0 : FL=0                          <29>
610 RETURN                               <242>
  ENDE DES LISTINGS
   [DRUCK:F.B.]
```

MERGE-ROUTINE für den COMMODORE C 16/116

Ein weiterer oft schmerzlich vermißter Befehl ist „MERGE". Im Basic des C16/116 ist kein Befehl in dieser Art zu finden, mit dem man Basic-Programme aneinanderhängen kann. Dabei ist eine solche Option äußerst nützlich, wenn man sich zum Beispiel eine Anzahl häufig gebrauchter Basic-Unterprogramme, eine sogenannte Unterprogramm-Bibliothek, schreibt und für ein bestimmtes Programm nun diejenigen Unterprogramme, die man benötigt, an sein Programm anhängen möchte. Dies war bisher nur möglich, indem man die Unterprogramme jeweils eintippte. Hier leistet diese Merge-Routine Abhilfe. Sie ist vollständig in Maschinensprache geschrieben und sehr kurz gehalten - lassen Sie sich nicht durch die Länge des Basic-Loaders täuschen. Er ist nur dazu da, das Maschinenprogramm auch für den Basic-Programmierer anwenderfreundlich zu gestalten. Gehen Sie also folgendermaßen vor:

- Geben Sie den abgedruckten Basic-Loader ein und speichern Sie ihn vorsichtshalber ab.
- Starten Sie das Programm mit RUN.
  Datassettenanwender müssen vorher in Zeile 10150 die 8 in eine 1 ändern.
- Legen Sie nun wie aufgefordert die Diskette/Kassette ein und drücken Sie eine Taste.
- Wenn alles in Ordnung ist, wird nun das Maschinenprogramm, es ist wie gesagt extrem kurz, unter dem Namen „MERGE.CODE" abgespeichert.

Für die spätere Anwendung gehen Sie wie folgt vor.

- Laden Sie die Merge-Routine mit
  LOAD"MERGE.CODE",8,1
  bzw. bei Kassettenbetrieb mit
  LOAD"MERGE.CODE",1,1
  ein.
- Laden Sie direkt danach das erste der zu verknüpfenden Basicprogramme ein.
- Starten Sie die Merge-Routine mit SYS550,"prgname",8 bzw. bei Kassettenbetrieb mit SYS550,"prgname",1. Dabei müssen Sie natürlich für prgname den Namen des zweiten Basic-Programms einsetzen, das an das erste angehängt werden soll. Das zweite Programm wrd nun nachgeladen und mit dem ersten verknüpft. Theoretisch können dann weitere Teile angehängt werden.

Folgende Punkte sind bei der Anwendung zu beachten:

- Die zu verknüpfenden Programme dürfen zusammen natürlich nur so lang sein, wie freier Basicspeicher zur Verfügung steht.
- Nach dem Laden muß die Merge-Routine auch sofort gestartet werden. Zum einen steht sie im Basiceingabepuffer und würde durch lange Eingaben gelöscht, und zum anderen sind nach dem Laden einige Pointer vermurkst.
- Sämtliche Zeilennummern des zweiten, anzuhängenden Programms müssen höher sein, als die Zeilennummern des ersten Programms, da sie sonst nicht erkannt werden. Dies kann man sich jedoch auch zunutze machen, indem man an ein eigenes Programm einen Copyright-Vermerk mit niedrigeren Zeilennummern anhängt. Ein solcher Copyright-Vermerk ist ohne weiteres nicht mehr löschbar.

Der Basic-Loader enthält ein dokumentiertes Assemblerlisting der Maschinenroutine.

„CPU" sucht Beiträge für die Rubrik „Tips u. Tricks"! - Wir zahlen 120,- DM pro abgedruckte Seite!!

```
0 REM ************************************************************
1 REM *                                                          *
2 REM *    M E R G E - R O U T I N E  FUER DEN COMMODORE C16/116 *
3 REM *                                                          *
4 REM *    BELEGT WIRD DER SPEICHERBEREICH VON 550 BIS 583       *
5 REM *    GELADEN WIRD MIT    LOAD"MERGE.CODE",8,1 [,1,1]       *
6 REM *    GESTARTET WIR MIT   SYS550,"PROGRAMMNAME",8 [,1]      *
7 REM *                                                          *
8 REM *    WRITTEN 1985 BY  W A L D E M A R  R A A Z             *
9 REM ************************************************************
10000 DATA20,91,94:REM JSR $9491    ; AUF KOMMA PRUEFEN
10001 :
10002 DATA20,6B,A8:REM JSR $A86B    ; LOAD-PARAMETER HOLEN
10003 :
10004 DATA38       :REM SEC         ; STARTADRESSE FUER DIE
10005 :
10006 DATAA5,2D    :REM LDA $2D     ; LOAD - ROUTINE AUS DEM
10007 :
10008 DATAE9,02    :REM SBC #$02    ; EHEMALIGEN BASIC-ENDE
10009 :
10010 DATAAA       :REM TAX         ; ERRECHNEN UND DER
10011 :
10012 DATAA5,2E    :REM LDA $2E     ; LOAD - ROUTINE UEBER-
10013 :
10014 DATAE9,00    :REM SBC #$00    ; GEBEN
10015 :
10016 DATAA8       :REM TAY
10017 :
10018 DATAA9,00    :REM LDA #$00    ; FLAG FUER LOAD SETZEN
10019 :
10020 DATA20,D5,FF:REM JSR $FFD5    ; LOAD - ROUTINE
10021 :
10022 DATA20,18,88:REM JSR $8818    ; BASICZEILEN NEU BINDEN
10023 :
10024 DATA20,4B,88:REM JSR $884B    ; BASICPOINTER NEU SETZEN
10025 :
10026 DATA20,93,8A:REM JSR $8A93    ; SONSTIGE POINTER ANGLEICHEN
10027 :
10028 DATA4C,03,87:REM JMP $8703    ; ZURUECK IN DEN READY-MODUS
10029 :
10030 : FOR I = 550 TO 583
10031 :    READ A$
10032 :    A = INPUT(A$)
10033 :    P = P + A
10034 :    POKE I , A
10035 : NEXT I
10036 :
    10037 IFP=3550THEN10080
    10040 PRINT"■ IHNEN IST BEIM ABTIPPEN DER DATAZEILEN"
    10050 PRINT" EIN FEHLER UNTERLAUFEN.   UEBERPRUEFEN"
    10060 PRINT" SIE BITTE DIE ENTSPRECHENDEN PROGRAMM-"
    10070 PRINT" ZEILEN (10000-10037) !":END
    10080 PRINT"■ DIE DATAWERTE SIND RICHTIG EINGELESEN."
```

```
10090 PRINT" DAS PROGRAMM WIRD JETZT ALS REINE MA-"
10100 PRINT" SCHINENROUTINE ABGESPEICHERT."
10110 PRINT" LEGEN SIE EINE DISKETTE / KASSETTE EIN"
10120 PRINT" UND DRUECKEN SIE EINE TASTE !"
10130 POKE239,0:WAIT239,1
10140 :
10150 SYS43115 "MERGE.CODE",8 : REM BEI KASSETTENBETRIEB: ...,1
10160 POKE38,38:POKE39,2:POKE2035,84:POKE2036,2
10170 POKE2034,38:SYS65496
10180 PRINT"■ DAS PROGRAMM IST NUN ALS MASCHINEN-"
10190 PRINT" ROUTINE ABGESPEICHERT !"
```

OLD-ROUTINE für den COMMODORE C 16/116

- Wenn alles in Ordnung ist, wird die Routine jetzt automatisch abgespeichert. Wie gesagt ist sie extrem kurz, was sich angenehm in den Speicher- und Ladezeiten äußert.

Haben Sie sich nun ein Programm gelöscht, nehmen Sie das gerade generierte Maschinenprogramm (nicht etwa den Basic-Loader) zur Hand und laden ihn mit
LOAD"OLD.CODE",8,1 bzw. bei Kassettenbetrieb mit
LOAD"OLD.CODE",1,1
ein.
Sobald sie es dann mit SYS216 starten, wird Ihr Basic-Programm wieder regeneriert. Es ist nach der "Behandlung wie neu". Zu beachten ist noch, daß auf keinen Fall neue Basiczeilen eingegeben werden dürfen, nachdem man sein Programm versehentlich gelöscht hat. Dadurch würde das Programm endgültig gelöscht. Für Interessierte enthält der Basic-Loader noch ein Assemblerlisting der Maschinenroutine.
Trotz des recht komfortablen Basic des C16/116 ist der Befehlsvorrat keineswegs komplett. So fehlt zum Beispiel ein Befehl „OLD" oder „RENEW" oder wie er sonst in anderen Basicversionen genannt wird, mit dem man „gelöschte" Basic-Programme weder herstellen kann. Nach einem Reset oder dem versehentlichen Benutzen des Befehls „NEW" ist ein Basic-Programm nicht etwa gelöscht, sondern alle Pointer werden so gesetzt, daß es für den Computer so aussieht, als sei der Speicher leer. In Wirklichkeit aber befindet sich das Programm noch vollständig im Speicher, es muß nur wieder regeneriert werden. Dies erledigt meine Old-Routine.
Sie ist – lassen Sie sich durch die Länge des abgedruckten Basic-Loaders nicht täuschen – extrem kurz, nämlich ganze 17 Bytes lang, natürlich vollständig in Maschinensprache geschrieben und kann nach dem versehentlichen „Löschen" des Basic-Programms nachgeladen werden und muß nicht etwa aktiviert werden, bevor man mit dem Programmieren beginnt.
Abgedruckt ist der Basic-Loader der Maschinenroutine mit einer integrierten Speicherroutine, die es auch dem Basic-Programmierer ermöglicht, auf sehr einfache Weise das Maschinenprogramm abzuspeichern.
Gehen sie dazu wie folgt vor:

- Tippen Sie das abgedruckte Programm ein und speichern Sie es vorsichtshalber schon ab.
  Datassettenbenutzer müssen in Zeile 10150 die 8 durch eine 1 ersetzen.
- Startem Sie das Programm mit RUN.
- Legen Sie wie aufgefordert eine Diskette bzw. eine Kassette ein und drücken Sie eine beliebige Taste.

```
 0 REM ************************************************************
 1 REM *                                                          *
 2 REM *    O L D - R O U T I N E  FUER DEN COMMODORE C16/116     *
 3 REM *                                                          *
 4 REM *    BELEGT WIRD DER SPEICHERBEREICH VON 216 BIS 232       *
 5 REM *    GELADEN WIRD MIT    LOAD"OLD.CODE",8,1 [,1,1]         *
 6 REM *    GESTARTET WIRD MIT  SYS216                            *
 7 REM *                                                          *
 8 REM *    WRITTEN 1985 BY  W A L D E M A R  R A A Z             *
 9 REM ************************************************************
10000 DATAA9,01    :REM LDA #$01     ; DUMMY AN DEN BEGINN
10001 :
10002 DATAA8       :REM TAY          ; DES BASICSPEICHERS
10003 :
10004 DATA91,2B    :REM STA ($2B),Y  ; SCHREIBEN
10005 :
10006 DATA20,18,88:REM JSR $8818     ; BASICZEILEN NEU BINDEN
10007 :
10008 DATA20,4B,88:REM JSR $884B     ; BASICPOINTER NEU SETZEN
```

```
10009 :
10010 DATA20,93,8A:REM JSR $8A93    ; SONSTIGE POINTER ANGLEICHEN
10011 :
10012 DATA4C,03,87:REM JMP $8703    ; ZURUECK IN DEN READY-MODUS
10013 :
10014 : FOR I = 216 TO 232
10015 :    READ A$
10016 :    A = INPUT(A$)
10017 :    P = P + A
10018 :    POKE I , A
10019 : NEXT I
10020 :
10030 IFP=1492THEN10080
10040 PRINT"■ IHNEN IST BEIM ABTIPPEN DER DATAZEILEN"
10050 PRINT" EIN FEHLER UNTERLAUFEN.   UEBERPRUEFEN"
10060 PRINT" SIE BITTE DIE ENTSPRECHENDEN PROGRAMM-"
10070 PRINT" ZEILEN (10000-10012) !":END
10080 PRINT"■ DIE DATAWERTE SIND RICHTIG EINGELESEN."
10090 PRINT" DAS PROGRAMM WIRD JETZT ALS REINE MA-"
10100 PRINT" SCHINENROUTINE ABGESPEICHERT."
10110 PRINT" LEGEN SIE EINE DISKETTE / KASSETTE EIN"
10120 PRINT" UND DRUECKEN SIE EINE TASTE !"
10130 GETKEY G$
10140 :
10150 SYS43115 "OLD.CODE",8 : REM BEI KASSETTENBETRIEB: ...,1
10160 POKE38,216:POKE39,0:POKE2035,233:POKE2036,0
10170 POKE2034,38:SYS65496
10180 PRINT"■ DAS PROGRAMM IST NUN ALS MASCHINEN-"
10190 PRINT" ROUTINE ABGESPEICHERT !"
```

„FARB-IRQ"-Programm

Eine komfortable Anwendungsmöglichkeit für den VC-20 in allen Ausbauversionen

„Farb-IRQ" ist ein Anwendungsprogramm für den VC-20 in allen Ausbauversionen, d. h. es läuft sowohl mit 8 - oder 16 KRam Speicher - Erweiterung als auch in der Grundversion. Das Programm entstand aus dem Gedanken heraus, die recht komplizierte Farbeinstellung des Bildschirmfensters zu erleichtern. Das Programm läuft im Speicherbereich von 680 (hex. 02A8) - 767 (hex. 02FF). Da das Hauptprogramm noch recht viel Speicherplatz in diesem Bereich übrig ließ, entschloß sich der Autor noch die Routinen „Low - High - Byte" + „Garbage Collection" hinzuzufügen. Alle 3 Programme füllen den obengenannten Speicherbereich voll aus. Ein Basic-Loader ermöglicht das normale Einladen des Maschinenprogrammes. Eine kleine Erläuterung ist außerdem auch noch im Loader vorhanden.

Der erste Teil (eigentlich der Hauptteil) enthält den Farb-IRQ. Es belegt den Speicherbereich von 680 - 737. Anschließend folgt „Low - High - Bytes" von 738 - 764. Als letzter Teil belegt Garbage Collection den Speicherbereich von 765 - 767.

1. Farb-IRQ: Ermöglicht das Einstellen der Bildschirmfarben. Mit der „F1"-Taste stellt man die Farbe des Rahmens ein. Mit der „F7"-Taste hingegen die Farbe des Bildschirmhintergrundes. Durch drücken der „Run-Stop"- + „Restore"-Taste kann man diese Funktion abschalten. Zum Aktivieren genügt einfach SYS 680. Bitte beachten Sie, daß diese Funktion nur im Direktmodus aktiv ist; im Programmmodus hingegen nicht.
2. High - Low - Bytes: Diese Funktion zerlegt eine Zahl zwischen 0 - 65535 in ein sog. Low - und ein High - Byte. Aufgerufen wird diese Funktion mit SYS 738, ADRESSE. Das Programm gibt danach zuerst das Low - Byte und danach das High - Byte aus. Diese Daten werden durch einen Schrägstrich („/") unterteilt. Anwendungsbeispiel: Die Begrenzung des RAM-Speichers. Möchte man ein Maschinenprogramm oder einen neuen Zeichensatz im Speicher unterbringen, muß man sicherstellen, daß diese Daten nicht durch ein Basic-Programm überschrieben werden. Dazu begrenzt man meist das Ende des Speichers. In den Adressen 55 + 56 kann man dies einstellen. Dabei ist Adresse 55 für das Low - Byte verantwortlich, 56 hingegen für das High - Byte. Nehmen wir an, es soll ein Maschinenprogramm von 238 Bytes Länge am Ende des RAM-Speichers un-

tergebracht werden. Um die Anfangsadresse des Speicherbereichs herauszufinden, ab der das Maschinenprogramm geschützt werden muß, wird einfach Ende des RAM-Speichers - Programmlänge gerechnet. In der VC-20 Grundversion ist das Ende des möglichen RAM-Speichers 7679. Subtrahiert man nun die Maschinenprogrammlänge von 238 Bytes so erhält man die Adresse 7441. Ab dieser Adresse muß also das Maschinenprogramm einge„Poke"d werden und auch geschützt werden. Wir wollen nun die Adresse 7441 in ein Low - und in ein High - Byte zerlegen und anschließend die Adressen 55 + 56 Poken. Nach SYS 738, 7441 erhält man als Low - Byte die Zahl 17, als High - Byte die Zahl 29. Wir Poken als nun die Werte in die Speicherstellen 55 + 56 : POKE 55,17 : POKE 56,29. Nun ist das Maschinenprogramm geschützt und kann nicht durch ein Basicprogramm überschrieben werden. Dem Programmierer stehen somit viele Möglichkeiten offen. Als Ansporn zum Anwenden der „High - Low - Byte"-Methode, hier noch einige Adressen: 43 + 44 = Anfang des RAM-Speichers, 45 + 46 = Ende eines Basicprogrammes, gleichzeitig auch Anfang der Variablen ...Es gibt noch wesentlich mehr solcher Adressen. Als Literatur kann ich für den Interressierten das Buch „VC-20 INTERN" von Data Becker empfehlen.

3. Garbage Collection: Diese Funktion entfernt nach Aufruf durch SYS 765 alle nicht mehr gebrauchte Strings. Dadurch erhöht sich der freie Speicherplatz und es wird eine längere Garbage Collection vermieden. Besonders bei verstärkten String-Gebrauch kann eine Zwang - Garbage - Collection ausgelöst werden, die in den unglücklichsten Fällen bis zu 20 Minuten dauern kann. Das macht natürlich ein gutes Programm völlig ungebräuchlich. Durch zeitweisen Aufruf mit SYS 765 kann diese lange Wartezeit vermieden werden.

```
1 REM**********************************
2 REM*                                *
3 REM*    F A R B - I R Q  V 1.0      *
4 REM*                                *
5 REM* PROGRAMMED BY  ANDREAS FUHRIG *
6 REM*                                *
7 REM*          10. JULI 1985         *
8 REM*                                *
9 REM**********************************
10 POKE36879,8:PRINT"■";
15 PRINT"■■                              ■";:GOSUB1000:PRINT"■ ■       ■■FARB-IRQ■■
■ ■";
20 FORT=1TO18:GOSUB1000:NEXT:PRINT"■■                                   ■";
25 PRINT"■■■■■■■■■AN = ■SYS 680■"
30 PRINT"■■■AUS = ■STOP/RESTORE■":PRINT"■■■'F1' = RAHMEN":PRINT"■■■'F7' = HINTER
GRUND"
35 PRINT"■■■■■■■EINSTELLEN■"
40 PRINT"■■■BITTE TASTE DRUECKEN■":PRINT"■■■(C)'85 BY A.FUHRIG";:POKE198,0:WAIT1
98,1:POKE198,0
45 PRINT"■■■                             ■";:GOSUB1000:PRINT"■ ■    ■■HIGH-LOW-BYTES■■
 ■ ■";
50 FORT=1TO18:GOSUB1000:NEXT:PRINT"■■                                   ■";
55 PRINT"■■■■■■■■SYNTAX :■■":PRINT"■■■SYS 738 , ADRESSE"
60 PRINT"■■■AUSGABE :■■":PRINT"■■■■■LOW-BYTE■/■HIGH-BYTE■"
65 PRINT"■■■BEDINGUNGEN :■■":PRINT"■■■0=<■■ADRESSE■■K=65535"
70 PRINT"■■■BITTE TASTE DRUECKEN■":PRINT"■■■(C)'85 BY A.FUHRIG";:POKE198,0:WAIT1
98,1:POKE198,0
75 PRINT"■■■                             ■";:GOSUB1000:PRINT"■ ■ ■■GARBAGE COLLECTION■■
 ■ ■";
80 FORT=1TO18:GOSUB1000:NEXT:PRINT"■■                                   ■";
85 PRINT"■■■■■■■■DIESE FUNKTION      ■■■■ENTFERNT NICHTGE-■■■■■■BRAUCHTE STRINGS
!"
90 PRINT"■■■■SYNTAX :■■":PRINT"■■■SYS 765"
95 PRINT"■■■BITTE TASTE DRUECKEN■":PRINT"■■■(C)'85 BY A.FUHRIG";:POKE198,0:WAIT1
98,1:POKE198,0
100 RESTORE:FORT=680TO767:READA:POKET,A:P=P+A:NEXT
200 DATA120,169,181,141,143,2,169,2,141,144,2,88,96,165,157,240,10,165,203,201,3
9,240,7
300 DATA201,63,240,23,76,220,235,174,15,144,202,138,9,8,141,15,144,162,255,200,2
08,253
400 DATA202,208,250,240,233,173,15,144,24,105,16,208,233,32,253,206,32,107,201,1
66,20
500 DATA169,0,32,205,221,169,47,32,210,255,166,21,169,0,230,214,76,205,221,76,38
,213
600 DATA 11988 :REM -> PRUEFSUMME <-
800 READK:IFP<>KTHENPRINT"■■■DATA - FEHLER ■ERROR■":END
900 PRINT"■■■FARB-IRQ & LOW-HIGH   ■& GARBAGE COL. AKTIV":SYS680:NEW
1000 PRINT"■ ■                          ■ ■";:RETURN
```

Universelles Bussystem für VC-20 und C-64

Inbetriebnahme der Ein-und Ausgabekarte

1. Aufbau der Netzteilkarte

Auf der Netzteilkarte werden alle Spannungen, die für den Betrieb des gesammten Bussystems benötigt werden, erzeugt. An der Eingangsstiftleiste werden an Pin 3–5 und 6–8 20–24 VAC angelegt. Die Belastbarkeit sollte bei ca. 3Amp. liegen (diese Wechselspannung wird über die Pins 25–30 ebenfalls in den Bus eingespeist).
Nach der Gleichrichtung durch GL3 steht an Pin 22–24 eine ungeregelte Gleichspannung von 28–32 VOLT für den Bus zur Verfügung. Diese Spannung wird gleichzeitig zur Versorgung des Spannungsreglers IC4 benötigt. Damit stehen für den Bus an Pin 19 +15V zur Verfügung. Da auf der Netzteilkarte kein Platz für den Ladeelko zum Glätten der 30V vorhanden ist, sind zwei Elkos auf dem Bus (C1 + C2) montiert.
An Pin 24–26 und 28–30 der Eingangsstiftleiste werden 8–10V AC, ca. 3 Amp. benötigt. Diese Spannung wird durch GL2 gleichgerichtet und durch C8 geglättet, sowie mit IC5 auf +5V geregelt. Die Spannung, die an Pin 16+17 anliegt, speist alle Logikbausteine der Steckkarten.
Über die Kondensatoren C6+ C7 (Stromspiegel) wird dem Gleichrichter GL1 ca. 20V zugeführt. Am Ausgang von IC2 (Pin 20) stehen -15V zur Verfügung. Diese Spannung wird gleichzeitig dem Regler IC3 zugeführt. Am Pin 18 liegen -5V an.
Auf der Steckkarte befindet sich noch die Thermenierung. Bei größeren Bussystemen, wie hier z.B. mit 10 Steckplätzen, können unerwünschte Induktivitäten oder Kapazitäten auftreten, auch ein Beeinflussen der parallel verlaufenden Busleitungen bringt Probleme. Man kann diese Probleme aber ausschalten, indem man den Bus niederohmig macht. Dieses wird in diesem Fall durch die aktive Thermenierung erreicht. Der Vorgang geschieht folgendermaßen:
Am Poti P1, das in der Spannungsteilerkette R1+R2 zwischen +5V und -5V liegt, wird eine Spannung von 2,6V eingestellt. Diese Spannung wird dem Leistungsoperationsverstärker TCA 365 am Pin 1 (nicht invertierenden Eingang) zugeführt. Der Operationsverstärker ist als Impedanzwandler ohne Spannungsverstärkung geschaltet, d.h. die Spannung, die an Eingang 1 zugeführt wird, steht bis auf wenige mV Differenz Ausgang 4 zur Verfügung und wird über Pin 13 zum Bus geleitet. Auf dem Bus werden die Leitungen D0-D7, A0-A3 sowie S02 und CRW über die Widerstände R1-R14 mit der Thermenierung verbunden. Hierdurch werden die Leitungen niederohmig gehalten. Zum Einstellen der 2,6V geht man wie folgt vor:
Nach dem Einstecken der Netzteilkarte und dem Anlegen der zwei Wechselspannungen, mißt man die Spannung an der Buchsenleiste (auf dem Bus) BU5 Buchse Nr. 19 gegen OV Buchse Nr. 1. Man stellt jetzt mit dem Poti P1 auf der Netzteilkarte eine Spannung von +2,6V ein. Damit ist die Einstellung beendet.
Beim Belichten muß das Platinen-Layout spiegelbildlich auf der Fotoschicht aufliegen. Die Beschriftung darf nicht lesbar sein. Zum Bohren verwendet man 1 mm Bohrer. Die vier Löcher (mit X gekennzeichnet), werden mit 3,2 mm gebohrt. Sie dienen zur Befestigung des Kühlkörpers für IC5. Dieser Kühlkörper wird auf der Leiterbahnseite montiert.
Die Netzteilkarte wird wie auf dem Bestückungsplan ersichtlich, bestückt und gelötet. Die Drahtbrücken werden nach dem Einlöten so dicht wie möglich an der Platine abgeschnitten. An beiden Enden der Platine werden die 31 pol. Stiftleisten angelötet. Die drei Spannungsregler IC2-IC4 werden mit Kühlkörper montiert.

Um bei der Beschaffung der Bauteile Hilfen zu geben, folgender Tip: Ich habe alle Bauteile aus dem Conrad Katalog E 86 bestellt. Die Kühlkörper für IC2, IC3 und IC4 sind auf Seite 84, Best.-Nr. 18 84 17, Abmessung 28 x 24, 5 x 32 Wärmeableitung °C/W = 13. Für IC1 genügt ein kl. Blech, zur Not geht es auch ohne. Das IC1 findet man auf Seite 82, Best.-Nr. 17 87 48. Es ist aber auch mit dem Verstärker IC TDA 2002 oder ähnlichen möglich.

Für IC5 wird folgender Kühlkörper benötigt: Seite 85 Best.-Nr. 18 86 70, Abmessung 111 mm breit, 26 mm hoch, 75 mm lang, Wärmewiderstand 2,3°C/Watt.

Der Kühlkörper wird mit den Befestigungswinkeln so auf die Leiterbahnseite gedrückt, daß von der Bestückungsseite die vier mit X gekennzeichneten 3,2 mm Löcher auf die Befestigungswinkel des Kühlkörpers übertragen werden können. Danach werden die 4 Löcher mit 3,2 mm gebohrt.
Zuerst montiert man IC5 mit den Anschlüssen zur Platine. Die beiden Beine des Reglers werden 5 - 6 mm über den Kühlkörper abgeschnitten, danach werden 2 Drähte angelötet. An einer der zwei Befestigungsschrauben (Masse), wird ebenfalls ein Draht mittels Kabelschuh befestigt. Die Drähte sollten 1 -1,5 mm² haben.
Anschließend werden die Drähte, entsprechend dem Schaltplan, mit den Bohrungen A, B und C verbunden. Bei der Befestigung des Kühlkörpers ist darauf zu achten, daß der Kühlkörper keinen Kontakt mit den Leiterbahnen hat. Bei den Befestigungspunkten werden entsprechende Isolierdistanzscheiben zwischen Platine und Kühlkörper montiert.
Für die Stromversorgung sollten 2 Trafos mit den entsprechenden Spannungen und Strömen verwendet werden. Durch zwei getrennte Trafos wird bei impulsartiger Belastung eine Rückkopplung auf die +5V vermieden.

2. Test der Netzteilkarte in Verbindung mit der Buskarte

Wichtig: Beim Einstekken der Netzteilkarte in den Bus ist darauf zu achten, daß die Platine richtig herum eingesteckt wird. Bei den Bestückungszeichnungen wird darauf hingewiesen, welche Seite in den Bus gesteckt wird.

a) Die Buskarte, die über die Flachbandleitung mit dem Platinenstecker verbunden ist, wird in den Bus Steckplatz 1 eingesteckt. Dagegen wird die Netzteilkarte noch nicht eingesteckt.
b) Jetzt schaltet man den Rechner ein. Wenn sich der Rechner nach ca. 5 - 7 Sek. noch nicht auf dem Bildschirm gemeldet hat, wieder abschalten. In diesem Fall muß man noch einmal die gesamten Verbindungen kontrollieren.
Beim Ätzen können dünne Brücken zwischen den Leiterbahnen stehengeblieben sein. Ist der Fehler behoben (es sind normalerweise Kurzschlüsse) muß sich der

Rechner noch einmal melden - „keine Angst!" - kurzzeitige Unterbrechungen übersteht der Rechner klaglos. Es dürfen dabei jedoch keine Spannungen über 5V auf die Leitungen gelangen. Meldet sich der Rechner, ist vorerst alles o.k..

c) Jetzt steckt man die Netzteilkarte in den Bus und die Buskarte wird herausgezogen. Die zwei Trafos werden über vier Drähte mit einer 31-poligen Buchsenleiste verbunden, die anschließend auf der Eingangsstiftleiste aufgesteckt wird. Dann werden die Trafos eingeschaltet.

In Tabelle 1 ist zu sehen, welche Spannungen auf den Buchsenleisten 2-9 auf der Busplatine anliegen müssen.

Buchse 19 = ca. + 2,6 V
Buchse 20 = ca. + 5 V
Buchse 21 = ca. - 5 V
Buchse 22 = ca. + 15 V
Buchse 23 = ca. - 15 V
Buchse 24 = ca. 0 V
Masse

Verläuft die Prüfung positiv, ist der Test vorerst beendet.

3. Aufbau der Eingangskarte (20 Kanal)

Diese Karte dient zur Erkennung von Spannungspegeln, die extern angelegt werden und die der Computer erkennen soll.

Die Pegel für Low sollen zwischen 0 und ca. 2 Volt liegen. Für High sind ca. 2,8 - 100 Volt erlaubt. Der Minusanschluß der externen Spannung muß mit dem GND der Eingangskarte verbunden werden, z. B. Eingangspin 27/28.

Beim Aufbau der Eingangskarte muß keine bestimmte Reihenfolge eingehalten werden.

Für die IC sollten Sockel verwendet werden.

Für die Widerstände und Zehnerdioden reichen 1/4 Watt-Typen aus.

Beim Einstecken der IC in die Sockel auf richtige Polung achten.

Bei den IC Gehäusen befindet sich auf einer Seite in der Mitte eine kleine Kerbe oder eine kleine Vertiefung. Auf dem Bestückungsplan sind diese Kerben eingezeichnet. Die IC werden, wie beim Bestückungsplan angedeutet, eingesteckt.

Bei den Zehnerdioden ist ebenfalls auf richtige Polung zu achten.

Die Kathode wird durch einen Farbring gekennzeichnet.

Beim Bestückungsplan ist für ZD1 - ZD20 die vorgegebene Position des Farbringes gekennzeichnet.

Zur Schaltung:

IC1 ist die Schnittstelle zwischen Außenwelt und Computer. An die 4 Pin RS 0 - RS 3 werden die Adressleitungen A 0 - 3 angelegt. Mit diesen werden die 16 Register ausgewählt. Der Reseteingang Pin 34 arbeitet wie der Reset der CPU. Beim Einschalten des Computers oder von Extern (Reset-Taster) wird diese Leitung kurze Zeit Low. Ebenso wie der Computer wird IC1 initialisiert, d. h., die beiden Ports werden auf Eingang geschaltet. Die anderen Register werden ebenfalls so gesetzt, daß keine unkontrollierten Signale erzeugt werden (Timer Interupt Schieberegister usw.).

D0 - D7 sind die Datenleitungen. Über diese Leitungen werden die Daten von den beiden Ports zum Computer oder vom Computer zu den beiden Ports übertragen.

SO2 Pin 25 ist der Takt 2 der CPU. Die Frequenz beträgt knapp 1 Mhz.. Die ansteigende Flanke von SO2 signalisiert, daß die Daten auf dem Datenbus stabil sind und von IC1, falls es adressiert ist, in die Port A oder B geschrieben werden können.

CS1 und CS2 sind die Chipselekt Leitungen. Damit IC1 angesprochen werden kann, muß CS1 High- und CS2 Low-Pegel haben.

CS1 ist mit der Reset-Leitung verbunden, die im Normalfall immer High ist. CS2 ist mit der Enable-Leitung verbunden. Diese Leitung wird immer dann Low, wenn der Adressdecoder auf der Buskarte die einzelnen Steckplätze auswählt.

R/W Pin 22 ist die Schreib-/-Lese-Leitung.

Ist diese Leitung Low, so werden die Daten, die an Pin 26 - 33 anliegen in die Ausgangsports A oder B übertragen und zwischengespeichert. Ist diese Leitung High, so werden die Daten, die von außen am Port A oder B anliegen, an die Pin 26 - 33 übertragen, die mit dem Datenbus verbunden sind und vom Computer gelesen werden.

IRQ Pin 21 ist die Interruptleitung der CPU. Nachdem diese Leitung Low ist, wird, wenn vom Programm der Interrupt nicht abgeschaltet ist, der augenblickliche Befehl fertig abgearbeitet und dann an den IRQ-Vektor gesprungen. Der Interupt kann von IC1 erzeugt werden. Davon später mehr.

Sehen wir uns jetzt den Eingang PA0 an. Alle übrigen Eingänge arbeiten analog.

Der Eingang PA0 liegt am Pin 29 der Eingangsstiftleiste. Ist der Eingang offen oder unbeschaltet, so wird über Widerstand R 2.3 der Pin auf 0 Volt gelegt. Der Eingangswiderstand ist dadurch auf ca. 2 M/ohm festgelegt. Man kann den Widerstand je nach Bedarf verkleinern oder auch weglassen. Über R 1.3 wird die Eingangsspannung an den Eingang (3) von Gatter N3 angelegt.

Gatter N3 ist ein C= Mos IC vom Typ 4050. Es kann auch der Typ 4049 eingesetzt werden.

4049 invertiert die Eingangsspannung, 4050 gibt den Pegel unverändert weiter.

Alle Programme vom Autor wurden für das IC 4050 geschrieben. Wird jetzt eine Spannung = ca. 3 Volt angelegt, so wird der Ausgang von N3 und der Eingang PA0 von IC1 High. Die Zehnerdiode ZD3 vom Typ „BZX 83C 6V8" begrenzt bei Spannungen >6,8 Volt die Eingangsspannung für das Gatter N3 auf ca. 6,8 Volt +. Da der Widerstand R 1.3 einen Wert von 2,2 Megaohm hat, werden auch bei Spannungen >100 Volt die Ströme, die durch R 1.3 und ZD3 fließen auf unbedenklichen Grenzen gehalten. I = U / R gleich 0,00004236 Amper bei 100 Volt. Die Verlustleistung N = U x I gleich 0.004 Watt. Sollen Spannungen von 100 Volt angelegt werden, so sollte der Widerstand R 1.3 von 2 Widerständen von je 2.2 Megaohm ersetzt werden, die in Reihe zu schalten sind. Die normalen Widerstände haben nur eine sichere Spannungsfestigkeit von ca. 100 Volt, so daß ein Durchschlag im Widerstand stattfinden könnte.

Wird die Gleichspannung verkehrt angelegt, so wird von der Zehnerdiode die negative Spannung auf - 0.7 Volt begrenzt.

Für Wechselspannungen ist diese Eingangskarte nicht geeignet. Werden Wechselspannungen angelegt, so wird bei der negativen Halbwelle 0 Volt signalisiert.

Man kann sie aber zum Frequenzmessen verwenden. Bei 50 Hz würden 50 Impulse an die Portleitung PA0 angelegt.

Die Eingangskarte für Wechselspannung ist in Vorbereitung.

Achtung!!! Man sollte keine Spannung, die größer als 40 Volt ist, verwenden, da größere Spannungen gefährlich sind. In der Regel dürften 24 - 30 Volt genügen.

Sollen von außen nur Spannungen von 5 Volt oder TTL Pegel angelegt werden, so kann der Widerstand R 2.3 und die Zehnerdiode ZD3 weggelassen werden. Widerstand R 1.3 wird durch eine Drahtbrücke ersetzt.

Beim Test der Eingangskarte wird folgendermaßen verfahren: Die Eingänge für Port PA0 - PA7 und Port PB0 - PB7 werden parallel an eine regelbare Spannungsquelle angeschlossen, ebenso ein Spannungsmesser.

Jetzt wird Hilfsprogramm 1 eingegeben und gestartet (zuvor sollte natürlich die Buskarte, Netzteilkarte und die Eingangskarte in den Bus eingesteckt sein, der Rechner mit der Buskarte verbunden und die Netzteilkarte mit den 2 Wechselspannungen versorgt sein. Jetzt kann man die Spannungen an den Eingängen so weit erhöhen,

bis auf dem Bildschirm 2-mal die Zahl 255 erscheint. Wenn man jetzt noch ca. 0,2 Volt zugibt, so hat man auf der eigenen Karte den untersten Pegel für High. Auf die gleiche Weise testet man den Low Pegel. Man regelt die Spannung so weit herunter bis auf dem Bildschirm 2-mal die Zahl 0 erscheint. Jetzt noch ca. 0,2 Volt tiefer und man hat den höchsten Pegel für Low.

Dieses Programm eignet sich auch zur Fehlersuche z. B.:

Port	Pegel	=	Wert
Pao	1	=	1
Pa1	1	=	2
Pa2	1	=	4
Pa3	1	=	8
Pa4	1	=	16
Pa5	1	=	32
Pa6	1	=	64
Pa7	1	=	128

Um die für Wert stehenden Zahlen zu erzeugen, darf immer nur ein Eingang für Port A auf High Pegel liegen.

Werden mehrere Eingänge von Port A auf High Pegel gelegt, so addieren sich die Werte. Z. B. Pa2, Pa4, Pa5 auf High Pegel Anzeige = 4+16+32=52

Analog gilt dasselbe für Port B.

Achtung! - Variable A darf vom Hauptprogramm noch nicht benutzt sein, sonst ist das Hilfsprogramm auf eine freie Variable umzuschreiben!

Eingang IC 1	Wert
Ca1	2
Ca2	1
Cb1	16
Cb2	8

Diese Werte müssen für einen kurzen Moment auf den Bildschirm erscheinen.

Hilfsprogramm 2 dient zum Test der Handshake (Kontrolleitungen), das sind die Eingänge CA1 + CA2 und CB1 + CB2. Vom Hilfsprogramm 1 müssen die Zeilen 10-120 mit eingegeben werden.
Zeile 130 PCR Register wird so programmiert, daß ein positiver Impuls (ansteigende Flanke) an CA1 + CA2 und CB1 + CB2 die entsprechenden Bit im IFR Register setzt.
Zeile 150 IFR Register wird nach Variable C gerettet.

Zeile 22	Basisadresse BU2 für VC 20
Zeile 23	Basisadresse BU2 für VC 64
Zeile 26	Bei ungültiger Zahl neue Abfrage
Zeile 40	Variablen a wird die Basisadresse des Steckplatzes zugewiesen
Zeile 110	Port B von IC1 wird auf Eingang geschaltet
Zeile 120	Port A von IC1 wird auf Eingang geschaltet
Zeile 130	Port A wird in die Variable A1 geschrieben
Zeile 140	Port B wird in die Variable B1 geschrieben
Zeile 200	Die Werte von Port A und Port B auf den Bildschirm

Hilfsprogramm 1 kann in jedem Programm eingebaut werden. Dabei sollte Zeile 10-120 am Anfang des Programms stehen. Zur Abfrage der Ports wird mit Goto 130 in das Unterprogramm gesprungen. In Zeile 200 wird durch Goto zurückgesprungen. Die Werte für Port A + B stehen in den Variablen A1 für Port A und B1 für Port B.

Zeile 160-170 durch Lesen von Port A und B wird IFR Register gelöscht.
Zeile 200 IFR Register wird auf dem Bildschirm angezeigt.
Da im IFR Register die einzelnen Bit solange gesetzt bleiben, bis durch Lesen von Port A + B die Bit gelöscht werden, müssen nach jedem Abfragen des IFR Register das IFR Register wieder gelöscht werden, um einen erneuten Impuls erkennen zu können. Um negative Impulse erkennen zu können (abfallende Flanke) muß Zeile 130 Poke A + 12 , 0 lauten.

Dadurch kann man leicht einen Frequenzzähler für positive oder negative Impulse an den Handshakeleitungen realisieren.

18 Kanal Ausgangskarte TTL / Transistor

Die Ausgangskarte verleiht dem Computer die Fähigkeit, vom Programm aus Schaltungsaufgaben durchzuführen. Die Karte ist in der Lage TTL-Pegel zu liefern, oder durch einen Ausgangstransistor größere Ströme zu schalten. Wechselspannungen können nicht geschaltet werden. Eine Karte mit 8 Ausgangsrelais für 220 Volt und 8 Amper ist in Vorbereitung. Zusätzlich sind auf der Karte noch 2 Eingänge für die Handshake (Kontrolleitungen) CA1 und CB1 vorhanden.
Die Eingangsspannung und die Programmierung ist analog zu der 20 Kanal-Eingangskarte.
Werden Ausgangstransistoren verwendet, so sollte man die Transistoren mit den Anschlüssen so tief wie möglich in die Platine einstecken und anlöten. Der Transistor wird stehend eingelötet. Wenn man die Platine so legt, daß IC1 vom eigenen Blickwinkel wegzeigt, werden die Transistoren so eingesteckt, daß sich die Metallfahne auf der linken Seite befindet. Beim Bestückungsplan ist angegeben, wo Emiter, Basis und Kollektor angeordnet sein müssen.
Die Oberkante des Transistors sollte max. 20-22 mm Oberkante-Platine vorstehen. Da bei einigen Transistoren der Kollektor mit der Metallasche verbunden ist, wird dadurch verhindert, daß unzulässige Berührungen mit der Leiterbahnseite der nächsten Platine auftreten können.
Wer ganz sicher gehen will, kann auf der Oberkante der Metallfahnen einen schmalen Pertinaxstreifen mit wärmefestem Klebstoff ankleben.
Zur Schaltung:
IC1 ist wieder die Schnittstelle. Da das IC in der Eingangskarte schon beschrieben ist, wird hier und in Zukunft nicht mehr darauf eingegangen, außer im Falle der Programmierung. Wenn mit dem Bussystem gearbeitet wird, sollten immer zuerst die gesamten Verbindungen hergestellt werden. Dann wird das Bussystem eingeschaltet und zu guter letzt der Rechner.
Da beim Einschalten des Rechners die Resetleitung immer einen Low-Impuls abgibt, wird IC1 immer über den Reseteingang initialisiert. Wird in anderer Reihenfolge verfahren, so ist beim Einschalten des Bussystems die Resetleitung schon wieder High und die 16 Register werden willkürlich gesetzt; zuerst die beiden Eingänge CA1 und CB1 Eingangsstift 2 und 4.
Die Schaltung ist identisch mit der aus der Eingangskarte.
Betrachten wir jetzt den Ausgang PA0 Ausgangsstift 30 (die übrigen Ausgänge arbeiten analog): Nach dem Einschalten des Rechners ist PA0 auf Eingang geschaltet und PA0 liegt auf ca. 5 Volt+. Diese Spannung gelangt nach Gatter N1, wird dort invertiert, so daß der Ausgang von N1 auf Low liegt. Die Basis von Transistor T2 liegt somit auf weniger als 0,6 Volt und der Transistor ist gesperrt. Der Transistor ist als offener Kollektor geschaltet. Eine von einer seperaten Spannungsquelle gespeiste Lampe, Relais oder Motor ist in diesem Fall abgeschaltet. Die Minusleitung der seperaten Spannungsquelle muß mit dem GND (0 Volt) der Ausgangsstiftleiste verbunden werden, z. B. Stift 21, 19, 17, 15 usw.. Jeder GND kann bis ca. 2 Amper belastet werden.
Werden mehrere Verbraucher gleichzeitig geschaltet und liegt der Gesamtstrom über GND bei 9 Amper, so sollten 5 GND Stifte mit der Minusleitung der externen Spannungsquelle verbunden werden.

Soll jetzt der Ausgang eingeschaltet werden (Transistor T2 soll leiten), so muß PA0 auf Low gesetzt werden. Die Spannung wird vom Gatter N1 invertiert und am Ausgang erscheint High. Die Basis von T2 wird größer 0,7 Volt und der Transistor leitet. Vom Widerstand R1.2 wird der Basisstrom auf zulässige Werte begrenzt.
Da die meisten TTL-Gatter (N 1) besser eine Spannung auf Low ziehen als auf High treiben, unterstützt R2.2 das Gatter N1.
Wenn der Ausgang von N1 Low ist, muß das Gatter nur den Strom vom Widerstand R2.2 gegen Masse ableiten. Da R2.2 680 Ohm hat und ca. 4,5 Volt vernichtet werden müssen, fließt ein Strom von ca. 7 mA; ein zulässiger Wert für N1. Der Widerstand kann auch auf 1000 Ohm vergrößert werden. Wird der Ausgang von N1 High, kann der Strom von R2.2 zusätzlich zu N1 zur Basis von T2 fließen. Werden TTL Ausgänge gewünscht, so kann T2 und R1.2 weggelassen werden. Es muß eine Drahtbrücke vom Kollektor von T2 zur Eingangsseite von R1.2 gelötet werden.
Für die Gatter N1 - N18 können folgende Typen eingesetzt werden:

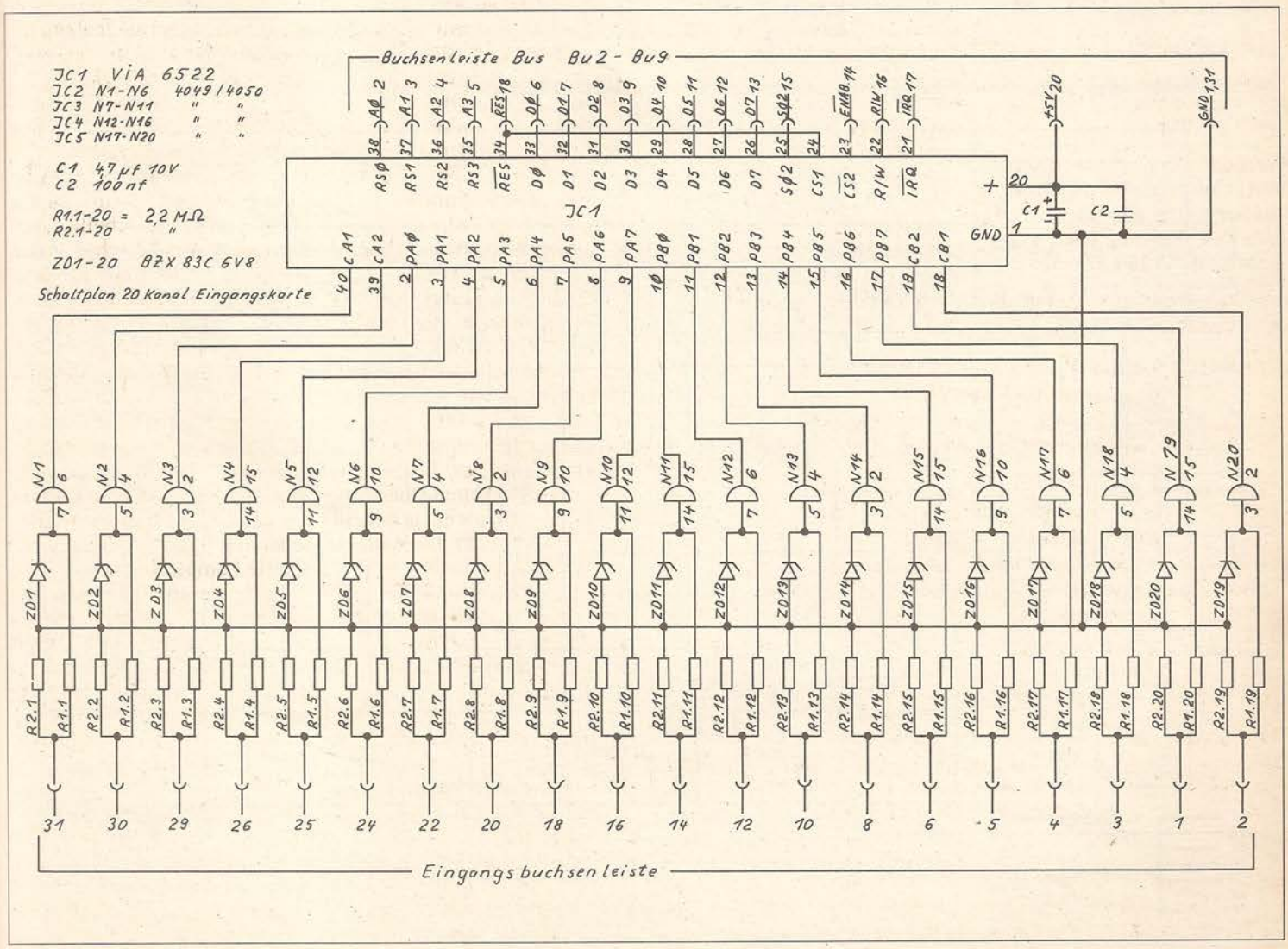
74LS04 = Inverter mit Gegentaktausgang
74LS05 = Inverter mit offenem Kollektor-Ausgang 5,5 Volt max.
74LS06 = Inverter mit offenem Kollektor-Ausgang 30 Volt max.
74LS07 = Treiber mit offenem Kollektor-Ausgang 30 Volt max.
74LS14 = Invertierende Schmitt-Trigger mit Gegentaktausgang
74LS16 = Inverter mit offenem Kollektor-Ausgang 15 Volt max.
74LS17 = Treiber mit offenem Kollektor-Ausgang 15 Volt max.

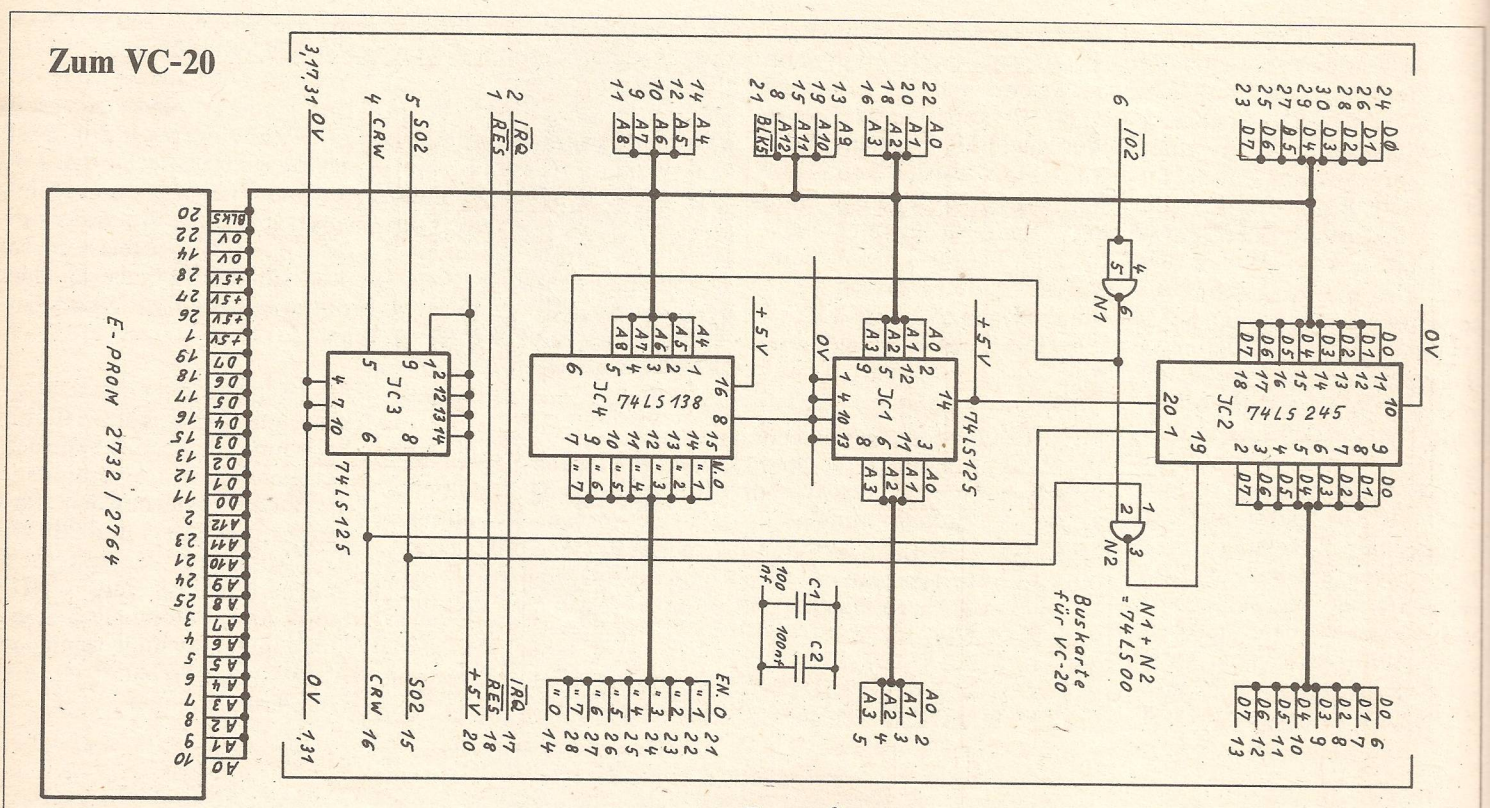
Der Autor hat die Typen 74LS04 eingesetzt. Die Anschlüsse und Innenschaltungen der genannten Typen sind identisch. Werden Transistoren eingesetzt, so werden NPN-Typen benötigt. Es sollten Darlingten Transistoren verwendet werden, da diese eine Spannung von größer 1 Volt an der Basis gegen den Emitter benötigen, um durchzuschalten.
Wenn Ausgang N1 Low wird (ca. 0,4 Volt), wird der Transistor sicher gesperrt.

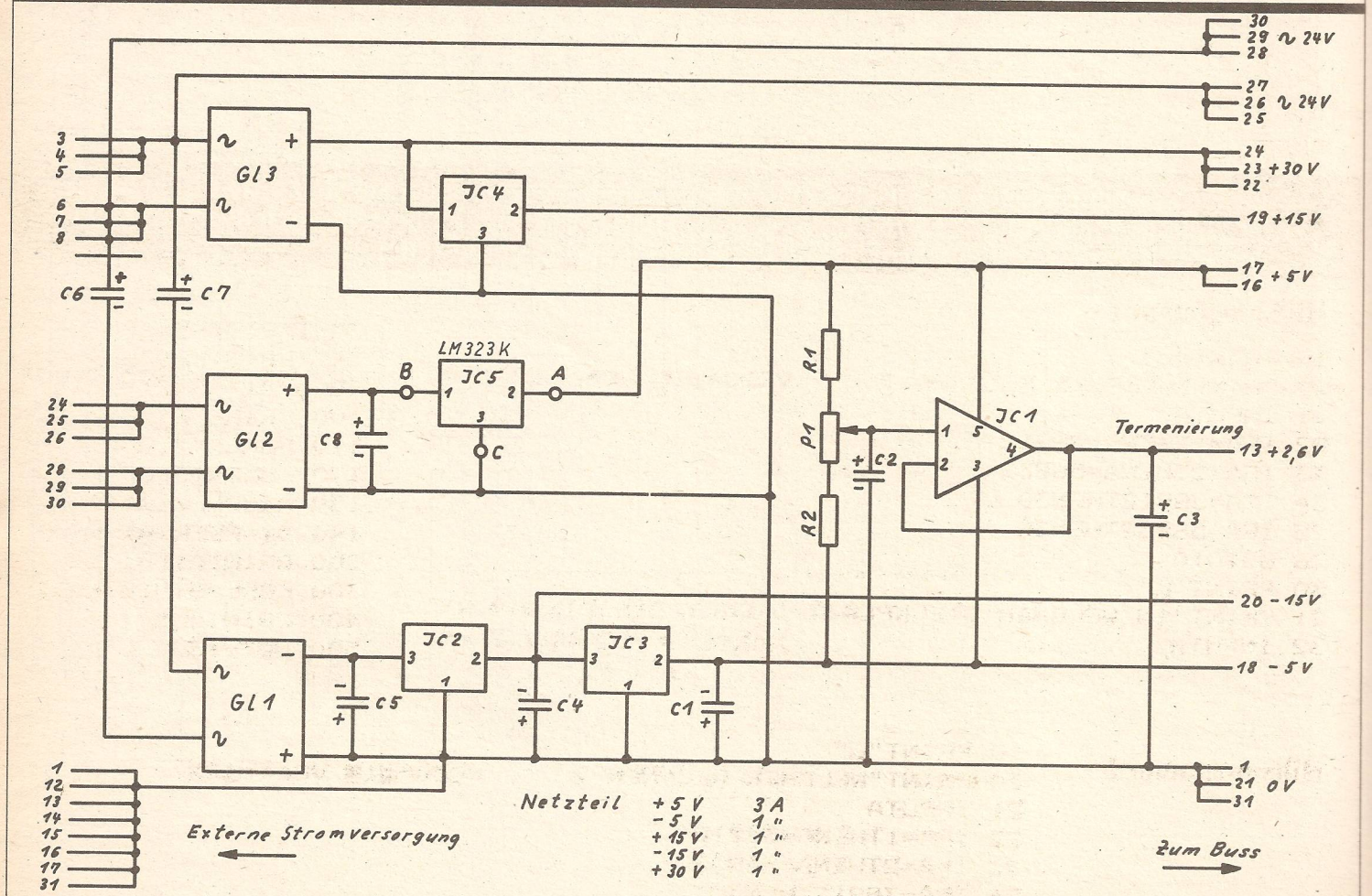
Es können folgende Typen verwendet werden:
BD 643, 645, 647, 649, 651, 675, 677, 679, 681, 683 usw.

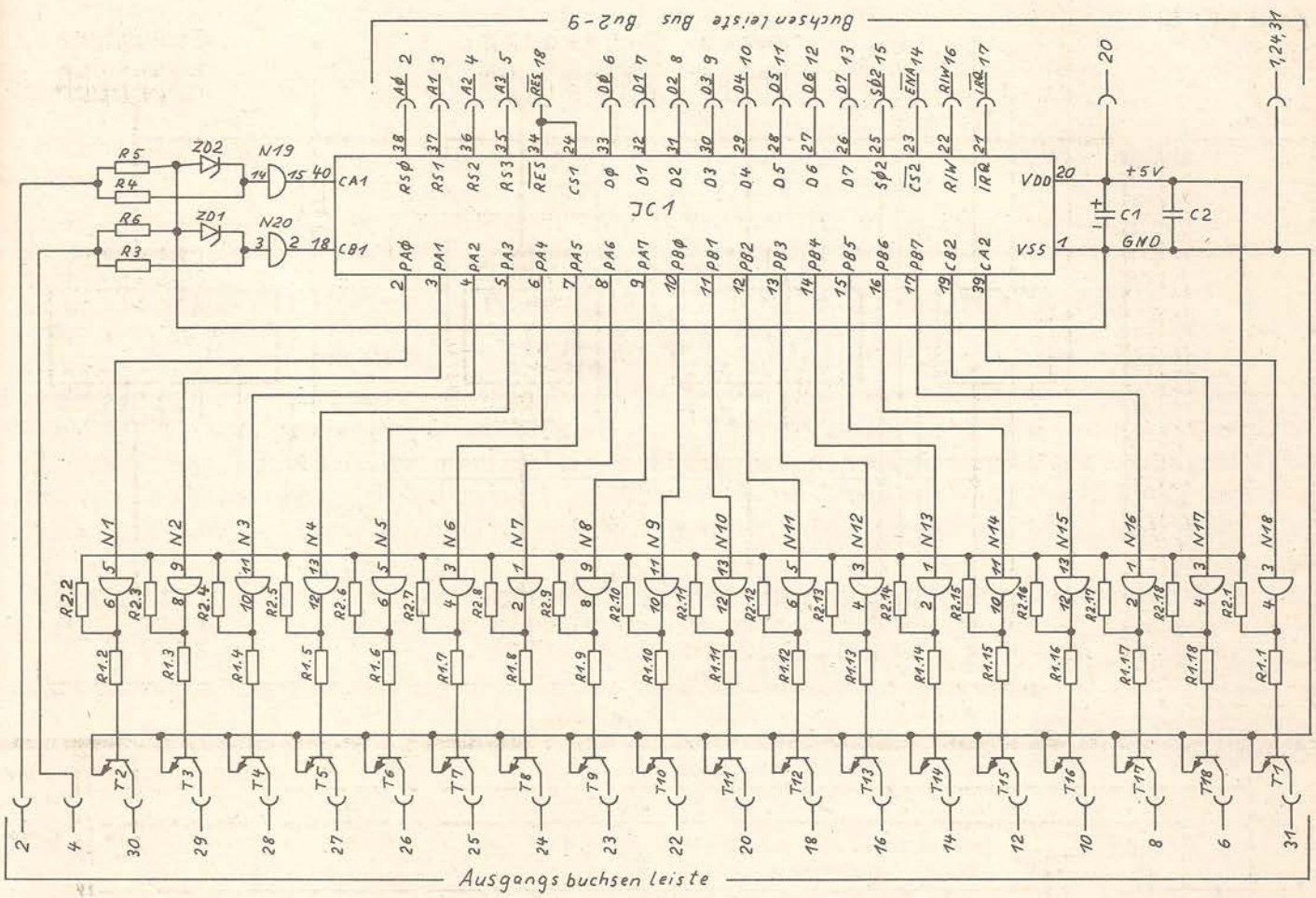
Der Autor hat den Typ BDW93C eingesetzt: Zum Test der Ausgangskarte Port A + B wird das Hilfsprogramm 3 eingesetzt. Werden die Transistoren T1 - T18 weggelassen, so muß für die Ausgabe an den Ausgangsstiften ca. 5 Volt liegen.

Sind Transistoren eingesetzt so kann man über eine externe Spannungsquelle eine Seite des Verbrauchers mit + verbinden, die andere Seite mit den Eingangsstiften PA0 - 7 und PB0 - 7, der Minuspol der externen Spannungsquelle mit GND. Sind die Verbraucher z. B. Glühlampen, so muß, wenn der Ausgang eingeschaltet ist, die Glühbirne leuchten.




Zum VC-20
E-PROM 2732 / 2764
74LS125
74LS138
74LS245
74LS125
74LS00
N1 + N2
Buskarte
für VC-20
Zum Bus


Gl3
Gl2
Gl1
LM323K
JC1
Termenierung
13 +2,6V
20 -15V
18 -5V
17 +5V
19 +15V
24 23 +30V
30 29 28 ~24V
27 26 25 ~24V
1 21 31 0V
Netzteil +5V 3A
-5V 1A
+15V 1A
-15V 1A
+30V 1A
Externe Stromversorgung
Zum Buss



Leser, die spezielle Fragen zu unserer Hardware-Serie haben, wenden Sie sich bitte direkt an den Autor Helmut Thöne (Tel. 0 56 76 / 13 76 - Di. u. Do. 18-19 Uhr). Es besteht auch die Möglichkeit bei Herrn Thöne die Platinen direkt zu bestellen.

Hilfsprogramm 1 -

```
10 PRINT"■"
20 PRINT"WELCHER RECHNER ?      VC20=■1■ VC64=■2■"
21 INPUTA
22 IFA=1THENA=38912
23 IFA=2THENA=56832
24 IFA=38912THEN30
25 IFA=56832THEN30
26 GOTO10
30 PRINT"■"
31 PRINT"IM WELCHEM STECKPLATZ STECKT DIE EINGANGS=
32 INPUTB                       KARTE ? BU2-BU9 2-9"
33 IFB>9THEN30
34 IFB<2THEN30
40 A=A+((B-2)*16)
100 PRINT"■"
110 POKEA+2,0
120 POKEA+3,0
130 A1=PEEK(A+1)
140 B1=PEEK(A)
200 PRINTA1;B1
300 FORL=0TO10:NEXT
400 PRINT"■"
500 GOTO130
```

Hilfsprogramm 2 -

```
10 PRINT"■"
20 PRINT"WELCHER RECHNER ?      VC20=■1■ VC64=■2■"
21 INPUTA
22 IFA=1THENA=38912
23 IFA=2THENA=56832
24 IFA=38912THEN30
```

```
25 IFA=56832THEN30
26 GOTO10
30 PRINT"␣"
31 PRINT"IM WELCHEM
STECKPLATZ STECKT
DIE EINGANGS KARTE ?
BU2-BU9 2-9"
32 INPUTB
33 IFB>9THEN30
34 IFB<2THEN30
40 A=A+((B-2)*16)
100 PRINT"␣"
110 POKEA+2,0
120 POKEA+3,0
130 POKEA+12,85
140 FORL=1TO500:NEXT
150 C=PEEK(A+13)
160 D=PEEK(A)
170 D=PEEK(A+1)
200 PRINTC
300 FORL=1TO200:NEXT
400 PRINT"␣"
500 GOTO140
```

Maschinensprachekurs für den ZX-Spectrum - Vom Byte zur Action - Teil 1

Maschinensprache - für Viele ein Zauberwort für unglaublich schnelle Programmabläufe, für fast ebenso Viele leider immer noch ein Buch mit sieben Siegeln. Dabei ist es, wenn man sich erst einmal mit der Materie vertraut gemacht hat, gar nicht einmal so schwer, effektive Programme in Maschinensprache zu schreiben. Lassen Sie sich also nicht entmutigen, wenn sie zu Beginn vor lauter Bäumen den Wald nicht mehr zu sehen glauben.
Das Ziel dieses Kurses ist es, einige grundlegende Dinge darzustellen, ohne allerdings zu tief in „graue Theorie" vorzudringen, um dann mit Hilfe dieser Grundlagen einen Weg zu zeigen, Maschinensprache vor allem für schnelle Spiele nutzbar zu machen.

I.1 Zahlensysteme

Das uns vom tagtäglichen Umgang her vertraute Zahlensystem ist das Dezimalsystem. Wie der Name schon besagt, beruht es auf der Zahl Zehn (lat. decem = zehn). In diesem System gibt es zehn „Zustände", nämlich die Ziffern von 0 bis 9. Sämtliche Zahlen des Dezimalsystems beruhen entweder auf diesen Ziffern selbst, oder auf einer Kombination derselben.
Der Computer ist, im Gegensatz zum Menschen, nicht in der Lage, zehn Zustände zu unterscheiden, sondern nur zwei, dargestellt durch die Ziffern 0 und 1.
Die Schwierigkeit besteht nun darin, daß wir uns vorerst aus einer Kombination der Ziffern 0 und 1 kein Bild der damit dargestellten Zahl machen können, einfach deshalb, weil wir normal eben nur im Dezimalsystem denken und rechnen. Zu vergleichen ist dieser Umstand mit dem Problem unterschiedlicher Währungen; mit einem gewissen Betrag Lire können wir auch erst etwas anfangen, wenn wir wissen, daß 100 Lire rund 1,50 DM wert sind.
Wir müssen uns also bemühen, die Kombination der Zahlen 0 und 1 zu „übersetzen", und zwar in das uns geläufige Dezimalsystem.
Doch zuerst wollen wir diesem aus zwei Zuständen (0 und 1) bestehenden System einen Namen geben: das Dualsystem (lat. duo = zwei).
Gezählt wird im Dualsystem folgendermaßen:

Dualsystem	Dezimalsystem
0	0
1	1
(den Zustand „2„ gibt es nicht, deshalb:)	
10	2
11	3
100	4
101	5
110	6
111	7
1000	8
1001	9
.	.
.	.
.	.

Übung:
Zählen sie auf diese Weise weiter bis zu der Zahl, der im Dezimalsystem die Zahl 29 entspricht.
Lösung:
(Der Zahl 29 entspricht die Dualzahl 11101)

Sicher werden Sie nun einwerfen, daß es etwas umständlich ist, den Wert einer Dualzahl durch Zählen herauszubringen. Es gibt aber eine relativ einfache Methode, den Wert einer Dualzahl auszurechnen. Um uns besser einzudenken, betrachten wir zuerst einmal wieder das uns bekannte Dezimalsystem.
Gehen wir dabei zum Beispiel von der dreistelligen Zahlenkombination 583 aus.
Die erste Stelle, die hier mit der Ziffer „3" belegt ist, gibt die Anzahl der „Einer", die zweite Stelle, hier mit der „8" belegt, gibt die Anzahl der „Zehner", die Ziffer „5" gibt schließlich die Anzahl der „Hunderter" an.
Wollte man den Wert dieser Zahlenkombination streng nach Schema berechnen, müßte man so vorgehen:
3 x 1 + 8 x 10 + 5 x 100 = 583.
Einer Zehner Hunderter

Natürlich ist es in der Praxis nicht nötig, eine Zahlenkombination des Dezimalsystems dermaßen zu zerlegen, da ihr Wert ja sowieso klar ersichtlich ist; wollen wir uns aber ein Bild von einer Dualzahl machen, ist diese Zerlegung unumgänglich.
Auch bei der Berechnung einer Dualzahl wollen wir zuerst von einem Beispiel ausgehen, und zwar von der dreistelligen Dualzahl-Kombination 110.
Die erste Stelle, hier mit der Ziffer „0" belegt, gibt die Anzahl der „Einer" an, die zweite Stelle, hier eine „1", die Anzahl der „Zweier", die dritte Stelle, ebenfalls mit „1" belegt, die Anzahl der „Vierer".
In unserem Falle berechnet sich die dazugehörige Dezimalzahl wie folgt:
0 x 1 + 1 x 2 + 1 x 4 = 6
Einer
Zweier
Vierer
Der Wert der Dualzahl 110 beträgt im Dezimalsystem also 6.
Nun kommen wir auf den Grundwert der einzelnen Stellen zu sprechen (im obigen Fall wäre der Grundwert bei den „Vierern" 4, bei den „Zweiern" 2, bei den „Einern" 1). Dazu ist zu sagen, daß jede Stelle gerade den doppelten Grundwert ihrer Vorgängerstelle besitzt. In unserem Beispiel hätte die nächste Stelle also den Grundwert 8

(2 x 4), die übernächste 16, dann käme 32 usw..
Da dies vielleicht zunächst ein bißchen verwirrend ist, gleich dazu ein Beispiel:
Nehmen wir die 6-stellige Zahlenkombination 101001.
Wir schlüsseln die Zahl gleich in einzelne Stellen auf und schreiben den Grundwert der Stelle gleich darunter.

Dualzahl	1	0	1	0	0	1
Grundwerte	32	16	8	4	2	1

Die Dezimalzahl berechnet sich jetzt also wie folgt:

$1 \times 1 + 0 \times 2 + 0 \times 4 + 1 \times 8 + 0 \times 16 + 1 \times 32 = 41$

Die Dezimalzahl zur Dualzahl 101001 beträgt also 41.
Noch ein Beispiel: 01010100

Dualzahl	0	1	0	1	0	1	0	0
Grundwerte	128	64	32	16	8	4	2	1

Also:

$0 \times 1 + 0 \times 2 + 1 \times 4 + 0 \times 8 + 1 \times 16 + 0 \times 32 + 1 \times 64 + 0 \times 128 = 84$

Aufgabe:
Berechnen Sie den Dezimalwert zu den folgenden Dualzahlen.

a. 10010111
b. 01010101
c. 10011001
d. 11011110
e. 00010010

Lösungen:
a. 151
b. 85
c. 153
d. 222
e. 18

Wenn Sie nun das Wesen der Dualzählweise verstanden haben, können wir zum nächsten Abschnitt weitergehen.

I.2 Bytes und Bits

Die einzelne Stelle einer Dualzahlen-Kombination nennt man im Fachjargon „Bit". Wenn man eine Kombination von acht solcher Bits vorliegen hat, spricht man von einem „Byte". 1024 solcher Bytes hingegen ergeben ein „Kilobyte". In einem Byte lassen sich also sämtliche 8-Bit Zahlen von 00000000 bis 11111111 unterbringen, was dezimal den Zahlen von 0 bis 255 entspricht.

Vorab sei bemerkt, daß der Computer auch die Möglichkeit besitzt, zwei Bytes „zusammenzuziehen", und damit Zahlen zwischen 0000000 000000000 und 1111111111111111, (dezimal zw. 0 und 65535) zu behandeln.

Auf diese Fähigkeit wird aber später noch gesondert eingegangen werden.

I.3 Der Aufbau eines Computers

Auch für den Programmierer, der sich hauptsächlich mit der Erstellung von „Software" beschäftigt, ist es unumgänglich, einige grundlegende Informationen über den Aufbau, die „Hardware" des Computers zu besitzen.

Das „Herz" eines jeden Computers ist der Mikroprozessor, kurz CPU genannt. Dieser Mikroprozessor hat die Aufgabe, Maschinensprachbefehle, die ihm aus einem Speicher zugeführt werden, zu lesen bzw. zu interpretieren und auszuführen.

Für diesen Vorgang stehen zwei Speicher zur Verfügung:
das RAM und das ROM (Abb. 3.1).

Das ROM wurde bereits vom Hersteller programmiert und dient zur Verwaltung von Basic-Befehlen bzw. -Programmen. Weil dieses ROM demzufolge eminent wichtig ist, kann es auch nicht gelöscht oder überschrieben werden. Im Gegensatz zu diesem ROM ist das RAM frei programmierbar und bietet daher die Möglichkeit, Maschinensprachprogramme aufzunehmen, die sich dann jederzeit abrufen lassen und von der CPU bearbeitet werden.

Eine besondere Art von Speichern steht uns noch in Form der sogenannten Register zur Verfügung. Uns interessieren zunächst einmal nur die 8 wichtigsten, und zwar die Register A,F,B,C,D,E,H,L. Sie befinden sich nicht wie Speicher RAM und ROM außerhalb der CPU, sondern sind vielmehr Bestandteil derselben (Abb. 3.2). Sie dienen dazu, Zahlen und Zwischenergebnisse zu speichern. Diese Register umfassen ein Byte (8 Bit), und sind deshalb gerade groß genug, Zahlen zwischen 0 und 255 aufzunehmen. Da dies aber oftmals nicht ausreicht, wandte man hier einen Trick an, und faßte die Register B und C, D und E sowie H und L zu den 16-bit Registern BC, DE und HL zusammen, so daß diese nunmehr jeweils Zahlen zwischen 0 und 65535 fassen können.
Besondere Bedeutung kommt den Registern A und F zu. Das „A" steht für die Bezeichnung „Akkumulator"; dieses Register ist für einfache Rechenoperationen wie Addition und Subtraktion verantwortlich.
Das Flagregister „F" ist im Prinzip nichts anderes als eine Kombination einzelner Zustandsanzeiger, die entweder 0 oder 1 sein können, und einzeln abgefragt werden können. Die einzelnen Zustandsanzeiger werden vom Ausgang einer Operation im Akku beeinflußt. so kann man zum Beispiel testen, ob das Ergebnis einer Operation im Akkumulator null war oder nicht. Damit werden die Zustandsanzeiger zu wertvollen Entscheidungshilfen bei Sprüngen. Ihr praktischer Einsatz wird aber noch genauer erläutert (Abb. 3.3).
Zu bemerken wäre noch, daß im vergangenen Abschnitt absichtlich auf tiefergreifende Erklärungen verzichtet wurde, der Sinn lag vielmehr darin, einige notwenige Grundbegriffe zu erwähnen.

Teil 2 unseres Maschinensprachkurses folgt in der nächsten CPU-Ausgabe

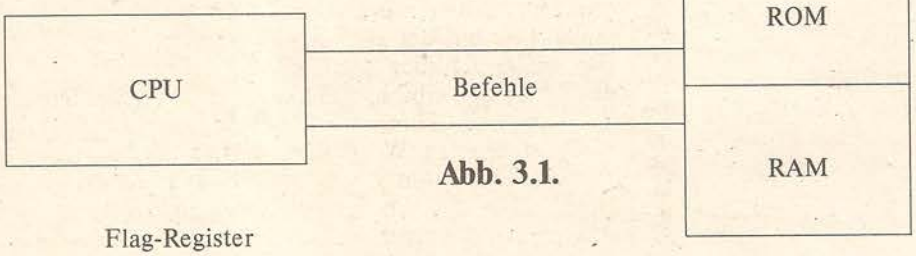


Abb. 3.1.

von den 8-Bits sind für uns nur zwei von Bedeutung:
Das Z-Flag und das C-Flag.

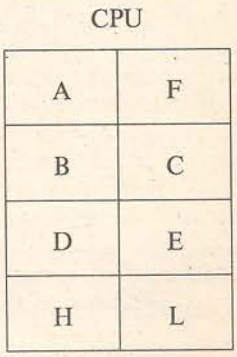


Abb. 3.3.

Martin Riedmiller

TIPS & TRICKS ❖ TIPS & TRICKS ❖ TIPS & TRICKS ❖

Drucker-Tips für den VC-20

A. Seitenanfang einstellen.

Mit folgenden Zeilen läßt sich der Drucker für die erste Zeile einer Druckseite einstellen:

```
10 OPEN4,4:B$="*****"
20 FOR I=1 TO 66
30 PRINT4$,I;B$
40 NEXT
50 CLOSE4
```

Das so erzeugte Blatt hat 66 Zeilen mit dem Druck ***** in jeder Zeile: Mit diesen Zeilen läßt sich die Position der ersten Zeile auf dem Blatt ziemlich genau einstellen.

B. Um mehr Zwischenraum auf einem Listing zwischen den einzelnen Zeilen zu bekommen, braucht man nur eine Filenummer größer als 127 in dem OPEN Befehl zu benutzen:

z.B. OPEN128,4:CMD4:LIST
Beendet wird die Routine mit:
PRINT#128:CLOSE 128

C. Maschinensprache-monitor-Hardcopy.

Mit der folgenden Routine wird der Bildschirmausdruck des Maschinensprachemonitors auf dem Drucker ausgegeben:
OPEN4,4:CMD4:SYS nnnnn:-Print#4:CLOSE4
nnnnn steht für die Startadresse des Monitors, z.B. 24576 bei dem VC 1213 Maschinensprache-Modul.

D. Programmlauf-Hardcopy.

Um den Bildschirminhalt eines Programmlaufes auf den Drukker umzuleiten, gibt man folgende Zeile als erste des Programmes ein:
1 OPEN4,4:CMD4:GOTOnnnn [erste eigentliche Programmzeile]
Das Programm muß dann mit folgender Befehlsfolge enden:
xxxxx PRINT#4:CLOSE4
[xxxxx=Zeilennummer]

E. Einfacher Bildschirmdump zum Drucker.

Der normale Bildschirminhalt - nicht Grafik HiRes - läßt sich mit folgender Befehlsfolge auf den Drucker geben:

```
OPEN3,3:OPEN4,4:-
PRINT"crsr home";:
FOR I = 0 TO 505:GET#3,A$:-
PRINT#4,A$:
NEXT:CLOSE3:CLOSE4
```

Vorher sollte der Bildschirm mit einer Zeile wie:
100GETA$:IFA$=""THEN100
'eingefroren' werden.

Das folgende Programm für eine LoRes Bildschirmkopie ist schon etwas umfangreicher:

```
100 SI$=CHR$(15):BS$=CHR$(8):D=
[Startadresse BildRAM]:OPEN4,4
110 FOR A = D TO D+22
120 PRINT#4,SI$
130 B=PEEK(A)
140 IF B >-1 AND B< 32 THEN E$=CHR$(B+64)
150 IF B >31 AND B< 64 THEN E$=CHR$(B)
160 IF B >63 AND B< 96 THEN E$=CHR$(B+32)
170 IF B >95 AND B<128 THEN E$=CHR$(B+64)
180 PRINT#4,E$
190 NEXT
200 PRINT#4,B$
210 D=D+22: IF D > 484 THEN 230
220 GOTO 110
230 CLOSE4:END
```

F. VC 20 und MPS 801: Absturzgefahr!

Einige Versionen des MPS 801 vertragen sich mit dem VC 20 nicht: das liegt an einem falschen 'Handshake' bei den früheren Versionen des MPS 801. Hat man einen solchen Drucker erwischt, läßt sich das Problem beseitigen, indem man gleich nach Einschalten des VC 20 - sowieso nach Einschalten des Druckers, damit dieser vom Computer auch erkannt wird - folgende POKEs eingibt:
POKE 37160,0:POKE 37161,1
Damit ist das 'Handshake' richtig eingestellt.

G. Ersatzfarbbänder

für den VC 1515 sind normalerweise – und mit ziemlicher Wartezeit - nur von Commodore zu bekommen; der Preis ist auch ziemlich hoch. Ein preiswerter Ersatz ist ein Ersatzfarbband für den DMP 100 Drucker von Tandy, vor allem wenn eine TRS-Vertretung in der Nähe ist.

H. Drucker als Schreibmaschine

Mit folgendem Programm kann man den VC 1515 bzw. 1525 in eine Schreibmaschine mit Sichtschirm umwandeln:

```
10 OPEN 4,4
20 GET A$: IF A$="" THEN 20
30 IF A$ = "@" THEN 70
40 PRINT#4, A$;
   REM SEMIKOLON!
50 PRINT A$;
   REM SEMIKOLON!
60 GOTO 20
70 PRINT#4:CLOSE 4:END
```

I. VC 1520 Plotter als Gerät Nr. 4.

Der VC 1520 hat bekanntlich immer Gerätenummer 6; mit folgendem kurzen Programm läßt sich das ändern:

```
10 FOR L = 0 TO 27
20 READ A:POKE L+679,A:
   NEXT
30 DATA 169,178,141,26,
   3,169,2,141,27,3,96,
   8,72,165
40 DATA 186,201,4,208,4,
   169,6,133,186,10,4,40,
   76,74,243
```

Aufgerufen wird die Routine mit SYS 679: wer die Routine nicht nach einem Warmstart ständig neu initialisieren will, kann sich die Maschinensprache vornehmen und die Routine ggfls. für den Stapelspeicher umschreiben

J. Bildschirmdump für den Superexpander.

Diese als Unterprogramm gedachte Routine erzeugt ein Hardcopy des mit dem Superexpander erzeugten HiRes-Bildes:

```
200 OPEN 4,4
201 FOR I = 7135 TO 4095
STEP -160
202 FOR H = 1 TO 160
205 TP = TP + 1
210 X = PEEK (I + H)
211 IF X >=128 AND X <192
    THEN X = X+64
212 IF X <=127 THEN X =
    X+128
215 A$ = A$ +CHR$(X)
220 IF TP = 160 THEN
    PRINT#4,CHR$(8) A$:
    A$ = "": TP = 0
230 NEXT H: NEXT I:
    PRINT#4: CLOSE 4:
    RETURN
```

Um einen doppelt breiten Ausdruck zu erhalten sind folgende Änderungen des Programms erforderlich:

```
213 IF TP <81 THEN A$ = A$
    + CHR$(X)
214 IF TP >80 THEN B$ = B$ +
    CHR$(X)
```

Zeile 220 wird ersetzt durch

```
225 IF TP = 160 THEN
    PRINT#$,CHR$(8)A$,B$:
    A$ = "": B$ = "": TP = 0
```

Dieses Programm läßt sich bei jedem HiRes-Bildschirm einsetzen, der aus 20 mal 20 Zeilen zu jeweils 8 mal 8 Pixels besteht; lediglich die Lage des BildschirmRAMs müßte in Zeile 201 angepaßt werden.

David Twigg-Flesner

Wer hat auch solche Tricks drauf? Schreiben Sie uns!

TIPS & TRICKS ❃ TIPS & TRICKS ❃ TIPS & TRICKS ❃

Tips und Tricks für den C-64:

„SCROLLER“

Das Scrollen des normalen Textbildschirms nach oben übernimmt das Betriebssystem des C-64, wenn man in der letzten Bildschirmzeile mit dem Cursor abwärts geht. Doch oftmals benötigt man einen Scrollvorgang nach unten. Dazu dient das Maschinenprogramm SCROLLER. Es verschiebt den ganzen Bildschirminhalt um eine Zeile nach unten und löscht den Inhalt der obersten Bildschirmzeile.

Das Programm ist als Basic-Lader gestaltet. Nach dem Starten wird die Maschinenroutine in den Speicher ab dez. 704 eingelesen, wobei eine Prüfsumme gebildet wird. Wenn diese stimmt, kann die Routine jederzeit mit SYS704 aufgerufen werden.

Da die Routine frei verschiebbar ist, kann sie auch in einen anderen Speicherbereich eingelesen werden. Dazu muß nur die Startadresse im Laderprogramm (N=704) entsprechend abgeändert werden. Die Routine kann selbstverständlich auch in eigene Programme eingebunden werden. Dann gestaltet man den Lader am besten als GOSUB-Routine, die man als erstes nach dem Programmstart anspringt.

Ralph Lässig

„Scroller“ – der BASIC-LOADER

```
100 REM  === SCROLLER      C64 ===
110 REM  (C)1985 BY RALPH LAESSIG
120 :
130 N=704:FORI=NTON+62
140 READQ:POKEI,Q:S=S+Q:NEXT
150 IFS<>9467THENPRINT"DATAFEHLER !!"
160 :
170 DATA169,192,133,251,169,007,133
180 DATA252,169,152,133,253,169,007
190 DATA133,254,162,024,160,039,177
200 DATA253,145,251,136,016,249,056
210 DATA165,251,233,040,133,251,176
220 DATA002,198,252,056,165,253,233
230 DATA040,133,253,176,002,198,254
240 DATA202,208,222,162,039,169,032
250 DATA157,000,004,202,016,250,096
```

ASSEMBLER-LISTING

```
110:    02C0                    .OPT P1,OO
120:    02C0                    *=   704
                        ;
                        ;
                        ;SCROLLER     C64
                        ;11.09.1985 BY RALPH LAESSIG
                        ;
                        ;
162:    02C0            VIDEO   =    1024
170:    02C0            START   =    1984
180:    02C0            ZEIGER1 =    251
190:    02C0            ZEIGER2 =    253
                        ;
220:    02C0 A9 C0              LDA  #<START
230:    02C2 85 FB              STA  ZEIGER1
240:    02C4 A9 07              LDA  #>START
250:    02C6 85 FC              STA  ZEIGER1+1 ;ZEIGER EINRICHTEN
```

```
260:   02C8 A9 98            LDA  #<(START-40)
270:   02CA 85 FD            STA  ZEIGER2
280:   02CC A9 07            LDA  #>(START-40)
290:   02CE 85 FE            STA  ZEIGER2+1
                ;
310:   02D0 A2 18            LDX  #24
320:   02D2 A0 27     L1     LDY  #39
330:   02D4 B1 FD     L2     LDA  (ZEIGER2),Y ;EINE ZEILE
340:   02D6 91 FB            STA  (ZEIGER1),Y ;VERSCHIEBEN
350:   02D8 88               DEY
360:   02D9 10 F9            BPL  L2
                ;
380:   02DB 38               SEC
390:   02DC A5 FB            LDA  ZEIGER1
400:   02DE E9 28            SBC  #40
410:   02E0 85 FB            STA  ZEIGER1
420:   02E2 B0 02            BCS  W1
430:   02E4 C6 FC            DEC  ZEIGER1+1
440:   02E6 38        W1     SEC            ;ZEIGER ERNIEDRIGEN
450:   02E7 A5 FD            LDA  ZEIGER2
460:   02E9 E9 28            SBC  #40
470:   02EB 85 FD            STA  ZEIGER2
480:   02ED B0 02            BCS  W2
490:   02EF C6 FE            DEC  ZEIGER2+1
                ;
510:   02F1 CA        W2     DEX
520:   02F2 D0 DE            BNE  L1        ;ZUR NAECHSTEN ZEILE
                ;
540:   02F4 A2 27            LDX  #39
550:   02F6 A9 20            LDA  #32
560:   02F8 9D 00 04  L3     STA  VIDEO,X ;OBERSTE ZEILE
570:   02FB CA               DEX
580:   02FC 10 FA            BPL  L3
                ;
600:   02FE 60               RTS
```

Tips und Tricks für den TI-99/4A

1. CALL LOAD (-31888,63,255)
Entspricht CALL FILES(Ø) und legt das Disketten-Betriebssystem lahm.

2. CALL LOAD(-31788,232)
Schaltet in den Multicolor-modus um.

3. CALL LOAD(-318Ø4,Ø,36)
Entspricht dem BYE-Befehl.

4. CALL PEEK(-28672,A)
Mit diesem Befehl kann man prüfen, ob der Sprachsynthesizer angeschlossen ist.
A=Ø Sprachsynthesizer nicht angeschlossen
A=96 Sprachsynthesizer ist angeschlossen

von
Holger Kusch

„GOTO X“:

Fortsetzung von Seite 37!

Tips und Tricks für den C-64!

Basic-Loader

```
100 REM  GOTO X
110 REM  (C)1985 BY RALPH LAESSIG
120 :
130 INPUT"STARTADRESSE";A
140 FORI=ATOA+11:READQ:POKEI,Q:NEXT
150 :
200 DATA032,253,174,032,138,173,032
210 DATA247,183,076,163,168
```

Das ASSEMBLER-LISTING:

```
110:    02C0                    .OPT P1,OO
120:    02C0                    *=   704
                          ;
                          ;GOTO X
                          ;30.08.1985  BY RALPH LAESSIG
                          ;
170:    02C0 20 FD AE           JSR  $AEFD   ;AUF KOMMA PRUEFEN
180:    02C3 20 8A AD           JSR  $AD8A   ;NUMERISCHEN WERT HOLEN
190:    02C6 20 F7 B7           JSR  $B7F7   ;INS ADRESSFORMAT WANDELN
200:    02C9 4C A3 A8           JMP  $A8A3   ;ZUM GOTO-BEFEHL
```

Korrektur zu Ausgabe 9: „Tips u. Tricks für SVI-328

1. CAPS LOCK

Der angegebene POKE funktioniert zwar, bringt den Rechner aber ziemlich durcheinander, wenn anschließend die CAPS LOCK-Taste betätigt wird. Der korrekte einzupokende Wert ist 32, nicht 1! Davon hätte sich der Autor leicht überzeugen können, wenn er einmal bei manuell eingeschaltetem CAPS LOCK unter der Adresse &HFE38 nachgesehen hätte!

Außerdem beeinflußt ein POKE nicht den Zustand der CAPS LOCK-LED in der Tastatur. Bevor hier jemand kommt und einen obskuren OUT-Befehl vorschlägt (der dann den Rechner zum Absturz bringt, wenn man mit der 64K-Erweiterung arbeitet), hier gleich der korrekte Weg:

- LED ein
OUT &H88,15:OUT &H8C, INP(&H90) OR 32

- LED aus
OUT &H88,15:OUT &H8C, INP(&H90) AND 223

2. Welche Taste?

Der „Druckfehlerteufel“ hat wieder zugeschlagen! Das hätte wohl eine Tabelle werden sollen. Nun ja, auch wenn man (in Gedanken oder durch Abschreiben) die Spalten gerade untereinanderrückt, ist es noch falsch. Außerdem fehlt jeder Hinweis, was denn die Spalten der Tabelle zu bedeuten haben. So wäre es richtig:
Jeder Taste ist also ein Bit zugeordnet; um eine bestimmte Taste abzufragen, ermittelt man aus der Tabelle zunächst die Speicherzelle, die gePEEKt werden muß, dann das abzufragende Bit.
Beispiel: PRINT-Taste
Aus der Zeile ergibt sich:
Speicher-Adresse &HFD7D
Aus der Spalte ergibt sich:
Bit 5 ≙ 32
Also:
IF (PEEK(&HFD7D) AND 32)=0 THEN PRINT “PRINT-Taste!“

Die beiden SHIFT-Tasten sowie die beiden ENTER-Tasten sind jeweils parallel geschaltet und lassen sich deshalb nicht unterscheiden.
Das gleichzeitige Drücken von mehr als zwei Tasten ist nur dann erlaubt, wenn alle in derselben Zeile oder Spalte liegen, sonst gibt’s “Geistertasten“!
Beispiel: Durch gleichzeitiges Drücken von Leertaste, Cursor rechts und Cursor abwärts wird die Funktionstaste F1 ausgelöst.

3. PLAY

Im “Programm“ fehlt wohl eine Zeile wie
110 IF PLAY(0) THEN 110

So sollte die Tabelle richtig aussehen!

	Bit							
Adresse	7 (128	6 64	5 32	4 16	3 8	2 4	1 2	0 1)
	Haupttastatur							
FD75	& 7	^ 6	% 5	$ 4	# 3	@ 2	! 1	> 0
FD76	? /	> .	+ =	< ,	" '	; :	(9	* 8
FD77	G g	F f	E e	D d	C c	B b	A a	. -
FD78	O o	N n	M m	L l	K k	J j	I i	H h
FD79	H w	V v	U u	T t	S s	R r	Q q	P p
FD7A	↑	⇦	{]	~ \	[	Z z	y y	X x
FD7B	←	ENTER	STOP	ESC	RIGHT GRPH	LEFT GRPH	CTRL	SHIFT
FD7C	↓	INS	HM CLS	F10 F5	F9 F4	F8 F3	F7 F2	F6 F1
FD7D	→		PRINT	SELECT	CAPS LOCK	DEL	⇨	Leertaste
	Zehnerblock							
FD7E	7	6	5	4	3	2	1	0
FD7F	,	=	/	*	-	+	9	8

4. CURSOR

Ich sehe keine sinnvolle Anwendung hierfür. Zumal der "Schattencursor" nur dann erscheint, wenn auch der richtige Cursor zu sehen ist; schaltet man diesen mit LOCATE,,0 aus, verschwindet auch sein zweites Abbild!

Uwe Schröder

TIPS & TRICKS für SVI-Besitzer - Fortsetzung aus Ausgabe 11/85

Schnellere Bildschirmausgabe

Viele Programme beinhalten eine Programmbeschreibung, die in Form von PRINT-Anweisungen auf den Bildschirm geschrieben wird, wie zum Beispiel Spielanleitungen oder allgemeine Informationen zur Programmbedienung.
Diesen Ausdruck möchte man in der Regel möglichst schnell erledigt haben.
Eine Möglichkeit, die PRINT-Anweisungen wesentlich schneller zu erstellen, ist die, vor den PRINT-Anweisungen den Cursor mittels LOCATE ,,0 auszublenden. Dies vermindert die Laufzeit um etwa 60 - 65 %!
Scheinbar noch schneller geht es mit folgendem Trick:
Man wählt die gleiche Vordergrund- wie Hintergrundfarbe. Dann wird der Text ausgedruckt, der allerdings noch nicht sichtbar ist. Nach der letzten PRINT-Anweisung wird die Vorder- bzw. die Hintergrundfarbe geändert. Auf diese Weise wird schlagartig der gesamte Text sichtbar.
Die beiden Tricks funktionieren allerdings nur im Textmodus (SCREEN 0).

Der Tastaturpuffer des SVI-328

Der Tastaturpuffer befindet sich im Bereich &H FD8B ... FDB2, umfaßt also 40 Bytes, das heißt es können maximal 40 Tastenbetätigungen zwischengespeichert werden. Dabei werden sämtliche Tastenbetätigungen der Reihe nach abgespeichert, also von &H FD8B beginnend bis &H FDB2 und dann wieder neu beginnend ab &H FD8B.
Wird zum Beispiel LIST ENTER eingegeben, so wird diese Eingabe im ASCII-Format der Reihe nach abgespeichert, nehmen wir einmal an von &H FD8B bis &H FD8F.
Nachdem der Puffer ausgelesen worden ist, werden die neuen Tastendrücke aber nicht wieder von der ersten Pufferstelle an abgespeichert, sondern ab &H FD90. Ist auf diese Weise zufällig einmal die Startadresse = &HFDB2, also die letzte Stelle im Pufferspeicher, so wird eben das erste Zeichen der Eingabe in &H FDB2 abgespeichert und die folgenden in &H FD8B, &H FD8C usw..
Damit der BASIC-Interpreter weiß, ab welcher Position die aktuellen Tasten gespeichert sind, benötigt er zwei Informationen:

a) Einen Speicher, der die Startadresse der Zeichen im Puffer angibt.
b) Einen Speicher, der entweder die Anzahl der Tastenbetätigungen angibt oder aber gleich auf das letzte aktuelle Zeichen im Tastaturpuffer hinzeigt. Letzteres ist im Prinzip beim SVI-328 der Fall.

Konkret hat dies das folgende Aussehen:
&H FA1C/FA1D : Zeiger auf Anfangsadresse
&H FA1A/FA1B : Zeiger auf Endadresse+1
Die Kenntnis darüber eröffnet nun eine Reihe ungeahnter Möglichkeiten. Als wichtigste sei die folgende herausgestellt:
In vielen Programmen sollen Funktionen bearbeitet werden, die der Benutzer erst während des Programmablaufs eingeben soll, zum Beispiel wenn mathematische Funktionen unterschiedlicher Form untersucht

werden sollen.
Die einzige Möglichkeit bestand bislang darin, die entsprechende Funktion vor Ablauf des Programms zu programmieren. Hatte man mehrere Funktionen zu untersuchen, bedeutete das also, jedesmal das Programm abzubrechen, um dann die neue Funktion zu programmieren und das Programm erneut zu starten.

Dies ist jedoch alles andere als benutzerfreundlich. Besser wäre es, dieses Problem durch eine einfache INPUT-Anweisung lösen zu können.

Doch jeder Versuch eines INPUT FNA oder INPUT A$:DEFFNA=A$ wird rigoros mit einer Fehlermeldung geahndet.

Wie man dennoch mittels INPUT eine Funktion definieren kann, soll nun ausführlich behandelt werden.
Möglich wird dies durch Manipulation des Tastaturpuffers. Hier gleich ein kleines Demonstrationsprogramm, das im Anschluß ausführlich besprochen wird:

```
10 ' Eingabe einer vom Benutzer definierten Funktion,
20 ' wie es v. a. bei Plott-Programmen erforderlich ist.
30 '
40 ' Die Funktion wird nachträglich in Zeile 150 programmiert.
50 CLS: LINEINPUT "FNA(X)=";A$
60 COLOR 1,1
70 CLS: PRINT"150 DEFFNA(X)=";A$
80 PRINT"GOTO 150"
90 POKE &HFA1C,&H8B
100 POKE &HFA1A,&H8E
110 POKE &HFD8B,11
120 POKE &HFD8C,13
130 POKE &HFD8D,13
140 END
150
160 COLOR 10,1 : LIST
```

Erläuterungen:
Zeile 50:
Eingabe der gewünschten Funktion in einer normalen Stringvariablen.
Zeile 60:
Die folgenden Bildschirmausgaben werden für den Benutzer unsichtbar durchgeführt; Vorder- und Hintergrundfarbe werden schwarz gewählt.
Zeile 70:
Hier wird die neue Zeile 150 so ausgedruckt, wie sie später im Programm stehen soll.
Zeile 80:
Erklärung folgt später.
Zeile 90:
Der Pointer auf die Startadresse des Tastaturpuffers wird auf &HFD8B gesetzt. Da das High-Byte des Pointers stets unverändert bleibt, genügt alleine die Änderung des LOW-Bytes in &HFA1C.
Zeile 100:
Analog zur Zeile 90: Der Pointer auf die Endadresse+1 wird auf &HFD8E gesetzt. Somit ergibt sich eine Länge von 3 Bytes im Tastaturpuffer. Diese werden in den
Zeilen 110-130
definiert: Das erste Zeichen ist ein CHR$(11) (Cursor Home), gefolgt von zweimal CHR$(13) (ENTER).
Zeile 140:
Nach Abbruch des Programms werden automatisch die „aktuellen" Zeichen des Tastaturpuffers ausgelesen und ausgedruckt, das heißt in diesem Falle: Der Cursor springt in die oberste linke Ecke, führt dann das erste ENTER durch, wobei die Zeile 150 ins Programm aufgenommen wird. Nach dem zweiten ENTER wird die Programmausführung mit
Zeile 150:
wieder aufgenommen. Jetzt wird also die Bedeutung von
Zeile 80:
klar! In
Zeile 160:
wird die Bildschirmdarstellung wieder sichtbar. Aus Demonstrationsgründen wird dann das Programm gelistet, um zu zeigen, daß wirklich eine neue Programmzeile im Programm vorhanden ist, ohne daß diese der Benutzer bewußt eingegeben hat.

Weitere Möglichkeiten für den Gebrauch des Tastaturpuffers lassen sich sehr schnell finden. Hier sind der Kreativität des Programmierers keine Grenzen gesetzt.

Joachim Pöhlmann

Ergänzung zum Programm „MD-Copierer" für den ZX-Spectrum

Nach dem Abtippen dieses Programmes erfolgte der erste Startversuch und wie erwartet stürzte das Programm ab. Die Erwartung eines Programmabsturzes unterliegt keinem Zweckpessimismus sondern trat nicht das erste Mal auf. Dies hat aber folgenden Grund. Sinclair hat das IF 1 ROM neu programmiert. Nicht nur die Fehler in der ersten Version wurden dabei beseitigt, sondern die Version 2 wurde teilweise ergänzt und neu strukturiert. Wer gerne wissen möchte, welche Version er besitzt, der sollte PRINT PEEK 23729 eingeben. IF 1 muß vorher initialisiert sein.
Ø = alte Version (ISSUE 1)
8Ø = neue Version (ISSUE 2
Daraus ergeben sich natürlich Probleme für die Programmierer. Benutzt man nämlich Unterroutinen im IF 1 ROM so kann es häufig vorkommen, daß die benutzten Routinen in den beiden Versionen an ganz anderen Stellen liegen. Dies war auch hier der Fall. Das vom Autor geschriebene Programm wurde auf einer ISSUE 1 Version entwickelt, ich dagegen besitze eine ISSUE 2 Version. Um dies einmal verdeutlichen zu können habe ich neben den Poke's auch einmal eine Vergleichstabelle aller in diesem Programm zu ändernden Unterroutinen angefertigt.
(Vergleichstabelle und Poke's siehe unter den abgedruckten Listings.)
Wer also Programme für beide Versionen entwickeln möchte, der kann zwei verschiedene Wege gehen.

1. Unterprogramme im IF 1 nur über RST 8, DEFB x aufrufen.
2. Für beide Versionen die unterschiedlichen Listings bzw. Poke's mit angeben.

In beiden Fällen ist jedoch eine genaue Lage aller Unterroutinen im IF 1 ROM wichtig. Die komplette Dokumentation hierüber ist in englischer Sprache erschienen:

Spectrum Shadow Disassembly
Gianluca Carri
Melbourne House Publishers
8.95 Pfund
in der BRD 39,80 DM

Hans-Jürgen Köhnen

Vergleichstabelle

ISSUE 1	BEZEICHN.	ISSUE 2
17F7	SEL-DRIVE	1532
12C4	GET-M-HD2	13A9
1E53	G-RDES	1E5E
12FE	RES-B-MAP	13E3
1D38	FREESECT	1D43
120D	WRITE-PRC	12EE
11A5	G-HD-RC	1280
1346	CHKS-BUFF	142B
12DA	CHK-MAP-2	13BF
182A	SW-MOTOR	1565
0700	UNPAGE	0700

```
          POKES ISSUE 2
63699,50        63700,21        64177,50        64178,21
63709,169       63710,19        64187,128       64188,18
63712,94        63713,30        64207,43        64208,20
63727,227       63728,19        64211,227       64212,19
64004,50        64005,21        64444,191       64445,19
64023,50        64024,21        64487,101       64488,21
64053,67        64054,29        64493,101       64494,21
64083,238       64084,18        64547,128       64548,18
64111,238       64112,18        64554,191       64555,19
                                64562,43        64563,20
```

Eingabehinweise für Commodore-C-64-Benutzer

Prüfsummen für Commodore 64
In den letzten Monaten wurde unsere Redaktion immer öfter auf sogenannte Prüfsummen angesprochen, die das Eingeben von Programmen erleichtern. Unser Spezialist Frank Brall hat daraufhin ein Verfahren entwikkelt, das das Eingeben von Commodore Programmen zum Kinderspiel macht.
Um die Eingabe unserer Listings zu erleichtern, werden zukünftig alle C64-Listings ohne Steuerzeichen abgedruckt. Die gesamten Steuerzeichen werden in unseren LISTINGS durch Klartext, der die Taste kennzeichnet, ersetzt. Eine Tabelle der Tastenfunktionen finden Sie in jedem Tronic-Magazin.

Folgendes Beispiel demonstriert die Arbeitsweise unseres Systems:

Hier muß ein Listing hergestellt werden!!!!!!!!!

Wie an diesem Beispiel zu erkennen ist, wird beispielsweise das Herz-Symbol durch das Wort CLEAR ersetzt. Dies bedeutet, Sie müssen die Taste CLEAR drücken um das Herz-Symbol zu erzeugen. Sollten Sie einmal nicht wissen welche Taste gemeint ist, so hilft Ihnen schnell unsere Tastentabelle, welche in jedem Magazin abgedruckt ist.
Um im Listing Tastenbezeichnungen von Grafik oder normalen Texten zu unterscheiden, werden alle Tastenkennzeichnungen in geschweifte Klammern gesetzt. Diese dürfen selbstverständlich nicht eingegeben werden.
Auch Leerzeichen innerhalb geschweifter Klammer dienen nur zur Trennung einzelner Tastenfunktionen und dürfen ebenfalls nicht eingegeben werden. Um die Tastenfunktionen noch besser hervorzuheben, werden diese in unterstrichener Kursivschrift (Schrägschrift) abgedruckt.
Steht hinter einer Tastenfunktion eine Zahl, die ebenfalls unterstrichen ist, so bedeutet dies, das die letzte Tastenfunktion mehrmals betätigt werden muß. Die in unserem Beispiel abgedruckte Funktion RIGHT3 bedeutet das die Taste RIGHT (Cursor rechts) 3 mal hintereinander betätigt werden muß. Auch einzelne oder auch mehrere Leerzeichen innerhalb von Anführungszeichen werden auf diese Art gekennzeichnet. Das bislang übliche Abzählen einzelner Zeichen entfällt somit völlig.
Alle Zeichen außerhalb der geschweiften Klammern werden normal abgedruckt und auch eingegeben.
Auf den ersten Blick sieht das sicher etwas kompliziert aus, ist jedoch in der Praxis ganz einfach. Wenn man sich erst einmal an die im Klartext geschriebenen Steuerzeichen gewöhnt hat, wird man den großen Vorteil dieser Schreibweise erkennen.

Checksummer C Version 1.0 für Commodore 64

Checksummer v1.0 ist ein Prüfsummen-Programm, das die Eingabe von Programm-Listings zum wahren Vergnügen macht. Tippfehler werden schon während des Eingebens der einzelnen Programmzeilen erkannt. Dieses System zusammen mit einem neuen Druckverfahren gewährleistet, daß unsere Listings zu 99% fehlerfrei abgedruckt werden.

Arbeitsweise und Aufbau unseres Checksummers:

Unser Checksummer besteht aus einem kleinen Maschinenprogramm, das als Basic-Loader abgedruckt ist.
Tippen Sie diesen Loader ein und speichern Sie ihn auf Kassette oder Diskette ab, denn Sie können ihn zukünftig immer wieder benutzen.
Der Start erfolgt durch den Befehl „RUN". Nach kurzer Zeit meldet sich der Rechner mit der Meldung „TRONIC . . .". Der Checksummer ist nun aktiv. Anschließend kann man ein beliebiges Tronic-Listing eingeben. Nachdem eine Zeile mit RETURN abgeschlossen wird, erscheint links oben auf dem Schirm eine Prüfzahl. Vergleichen Sie diese mit der Zahl, die im Heft hinter dieser Zeile abgedruckt ist. Stimmt die Zahl überein, so ist die Zeile richtig eingegeben, andererseits muß sich noch ein Fehler in Ihrer Eingabe eingeschlichen haben, und Sie müssen die Zeile korrigieren.
Auf diese Weise können Sie das gesamte Listing schnell und fehlerfrei eingeben.
Interessant ist auch, daß bei der Eingabe von Zeilen die üblichen Abkürzungen benutzt werden können, ohne die Checksumme zu verändern. Leerzeichen außerhalb von Anführungszeichen werden ignoriert, da diese auf die Ausführung der einzelnen Befehle keinen Einfluß haben. Sie können Ihr Programm auch starten, denn der Checksummer und Ihr Programm beeinflussen sich nicht gegenseitig. Wollen Sie den Checksummer abschalten, so geben Sie einfach POKE 1,55 ein oder betätigen die Tasten-Kombination RUNSTOP und RESTORE.
Aktivieren können Sie den Checksummer jederzeit (auch nach RESET) mit POKE 1,53.
Wer den Checksummer nicht eingeben möchte, kann diesen auch unter der folgenden Bestellnummer beziehen:
Bestellnummer:

CV10K Kassette 10,-- DM
CV10D Diskette 15,-- DM

Frank Brall

Tastenbezeichnungen für Steuercodes (TRONIC-NORM)

DOWN	CURSOR UNTEN	TASTE NEBEN RECHTEM SHIFT
UP	CURSOR HOCH	SHIFT-TASTE & TASTE NEBEN RECHTEM SHIFT
CLR	CLEAR SCHIRM	SHIFT-TASTE & 2. TASTE GANZ RECHTS OBEN
INST	EINFUEGEN	SHIFT-TASTE & TASTE GANZ RECHTS OBEN
HOME	CURSOR IN ECKE	2. TASTE VON GANZ RECHTS OBEN
DEL	DELETE	TASTE GANZ RECHTS OBEN
RIGHT	CURSOR RECHTS	TASTE GANZ RECHTS UNTEN
LEFT	CURSOR LINKS	SHIFT-TASTE & TASTE UNTEN RECHTS
SPACE	LEERZEICHEN	LEERTASTE (GROESSTE TASTE)
F1	FUNKTIONSTASTE	FUNKTIONSTASTE F1
F3	FUNKTIONSTASTE	FUNKTIONSTASTE F3
F5	FUNKTIONSTASTE	FUNKTIONSTASTE F5
F7	FUNKTIONSTASTE	FUNKTIONSTASTE F7
F2	FUNKTIONSTASTE	FUNKTIONSTASTE F2
F4	FUNKTIONSTASTE	FUNKTIONSTASTE F4
F6	FUNKTIONSTASTE	FUNKTIONSTASTE F6
F8	FUNKTIONSTASTE	FUNKTIONSTASTE F8
BLACK	SCHWARZ	CINTROL-TASTE & 1
WHITE	WEISS	CONTROL-TASTE & 2
RED	ROT	CONTROL-TASTE & 3
CYAN	TUERKIS	CONTROL-TASTE & 4
PURPLE	PURPUR	CONTROL-TASTE & 5
GREEN	GRUEN	CONTROL-TASTE & 6
BLUE	BLAU	CONTROL-TASTE & 7
YELLOW	GELB	CONTROL-TASTE & 8
RVSON	INVERSE EIN	CONTROL-TASTE & 9
RVSOFF	INVERSE AUS	CONTROL-TASTE & 0
ORANGE	ORANGE	COMMODORE-TASTE & 1
BROWN	BRAUN	COMMODORE-TASTE & 2
LIG.RED	HELLROT	COMMODORE-TASTE & 3
DGREY	DUNKELGRAU	COMMODORE-TASTE & 4
MGREY	MITTELGRAU	COMMODORE-TASTE & 5
LIG.GREEN	HELLGRAU	COMMODORE-TASTE & 6
LIG.BLUE	HELLBLAU	COMMODORE-TASTE & 7
HGREY	HELLGRAU	COMMODORE-TASTE & 8
CTRL	CONTROL	CONTROL-TASTE ZUSAMMEN MIT DEM NACHFOLGENDEN ZEICHEN. (Z.B. CTRLA ENTSPRICHT CTRL & A)

Achtung: Einige Listings enthalten das Zeichen "_", dieses Zeichen entspricht dem Pfeil nach links (Taste ganz links oben)

Der Tronic-Verlag steigt ein:

BTX-Software über Telefon!

Über 700.000 C64 sind bereits in Deutschland in Betrieb, jeder von Ihnen kann in Kürze durch einfaches Einstecken eines neu entwickelten Decodermoduls SD-64 zu einem Abrufterminal für Bildschirmtext umfunktioniert werden. Dieses Modul ist bisherigen, ungleich teureren Geräten nicht nur gleichwertig, sondern in Intelligenz und Funktion weit überlegen. Damit ist Bildschirmtext nicht mehr nur der professionellen Nutzung vorbehalten, sondern ein Medium für jedermann geworden.

Der Tronic-Verlag plant, in Zusammenhang mit dem Hersteller, ab 1986 einen „TELESOFT-Vertrieb" zu errichten. Dieser erlaubt das Abrufen von Qualitativ hochwertiger Software über Telefon. Damit ergibt sich endlich eine Möglichkeit preiswert und schnell Software zu vertreiben, denn Versand- und Bearbeitungskosten entfallen. Auch die Telefonkosten halten sich in Grenzen, da im gesamten Bundesgebiet zum Nahtarif telefoniert werden kann.

Wir werden in den kommenden Ausgaben noch näher auf das Thema und das neue preiswerte Modul eingehen.

Weitere Informationen von Mitarbeiter F. Brall

```
   C64 LISTING MIT CHECKSUMMEN (C V1.0)

0 REM ***** TRONIC CHECKSUMMER *****          <116>
1 REM VERSION C 1.0 BEI FRANK BRALL           <10>
2 REM -------------------------------         <148>
3 FOR I=832 TO 1008:READ A:S=S+A              <68>
4 POKE I,A:NEXT I:IF S<>16397 THEN PRINT
"FEHLER{SPACE}IN{SPACE}DATA-ZEILEN{SPACE
}!{SPACE}(10-19)":STOP                        <32>
5 SYS 832:S=0:FOR I=58464 TO 58603:READ
A:S=S+A                                       <21>
6 POKE I,A:NEXT I:IF S<>18919 THEN PRINT
"FEHLER{SPACE}IN{SPACE}DATA-ZEILEN{SPACE
}!{SPACE}(29-36)":STOP                        <219>
7 S=0:FOR I=48979 TO 49005:READ A:S=S+A       <33>
8 POKE I,A:NEXT I:IF S<>2888 THEN PRINT"
FEHLER{SPACE}IN{SPACE}DATA-ZEILEN{SPACE}
!{SPACE}(46-47)":STOP                         <23>
9 PRINT"{DOWN2}AKTIVIEREN{SPACE}:{SPACE}
POKE{SPACE}1,53{SPACE18}AUSSCHALTEN:{SPA
CE}POKE{SPACE}1,55":NEW                       <26>
10 DATA169,0,133,254,162,1,189,137,3,133
,255,160,0,177,254,145,254,136                <176>
11 DATA208,249,230,255,165,255,221,139,3
,208,238,202,16,230,169,96,141,49             <147>
12 DATA165,169,228,141,50,165,169,53,133
,1,169,141,133,254,162,3,134,255              <225>
13 DATA160,0,177,254,240,7,32,202,241,20
0,76,120,3,169,2,141,32,208                   <140>
14 DATA96,160,224,192,0,147,17,32,32,35,
35,35,35,32,84,82,79,78                       <158>
15 DATA73,67,45,86,69,82,76,65,71,32,67,
72,69,67,75,83,85,77                          <129>
16 DATA77,69,82,32,35,35,35,35,139,13,13
,32,32,32,32,32,32,32                         <147>
17 DATA32,32,32,32,32,86,69,82,83,73,79,
```

```
78,32,67,32,49,46,48                                    <79>
18 DATA13,13,32,32,32,32,32,32,40,67,41,
32,70,82,65,78,75,32                                    <94>
19 DATA66,82,65,76,76,32,40,49,48,46,56,
53,41,13,0                                              <12>
29 DATA169,0,141,107,191,141,108,191,141
,109,191,160,2,24,177,95,200,113                        <7>
30 DATA95,141,109,191,160,3,200,177,95,2
40,44,201,34,208,10,173,107,191                         <91>
31 DATA73,1,141,107,191,177,95,174,107,1
91,208,4,201,32,240,228,238,108                         <128>
32 DATA191,174,108,191,24,177,95,109,109
,191,141,109,191,202,208,244,76,120                     <244>
33 DATA228,56,76,183,228,72,32,201,255,1
70,104,144,1,138,96,32,240,255                          <101>
34 DATA142,105,191,140,106,191,162,0,160
,0,24,32,240,255,169,91,32,12                           <89>
35 DATA225,169,0,174,109,191,32,205,189,
169,93,76,83,191,173,33,208,145                         <239>
36 DATA243,96,105,2,164,145,200,208,4,19
7,161,208,247,96                                        <202>
46 DATA32,12,225,32,63,171,32,63,171,172
,106,191,174,105,191,24,32,240                          <213>
47 DATA255,76,128,164,24,0,0,5,200                      <65>
  ENDE DES LISTINGS
   [DRUCK:F.B.]
```

Der Atari 520 ST – *Ein Erfahrungsbericht von Klaus Löffelmann*

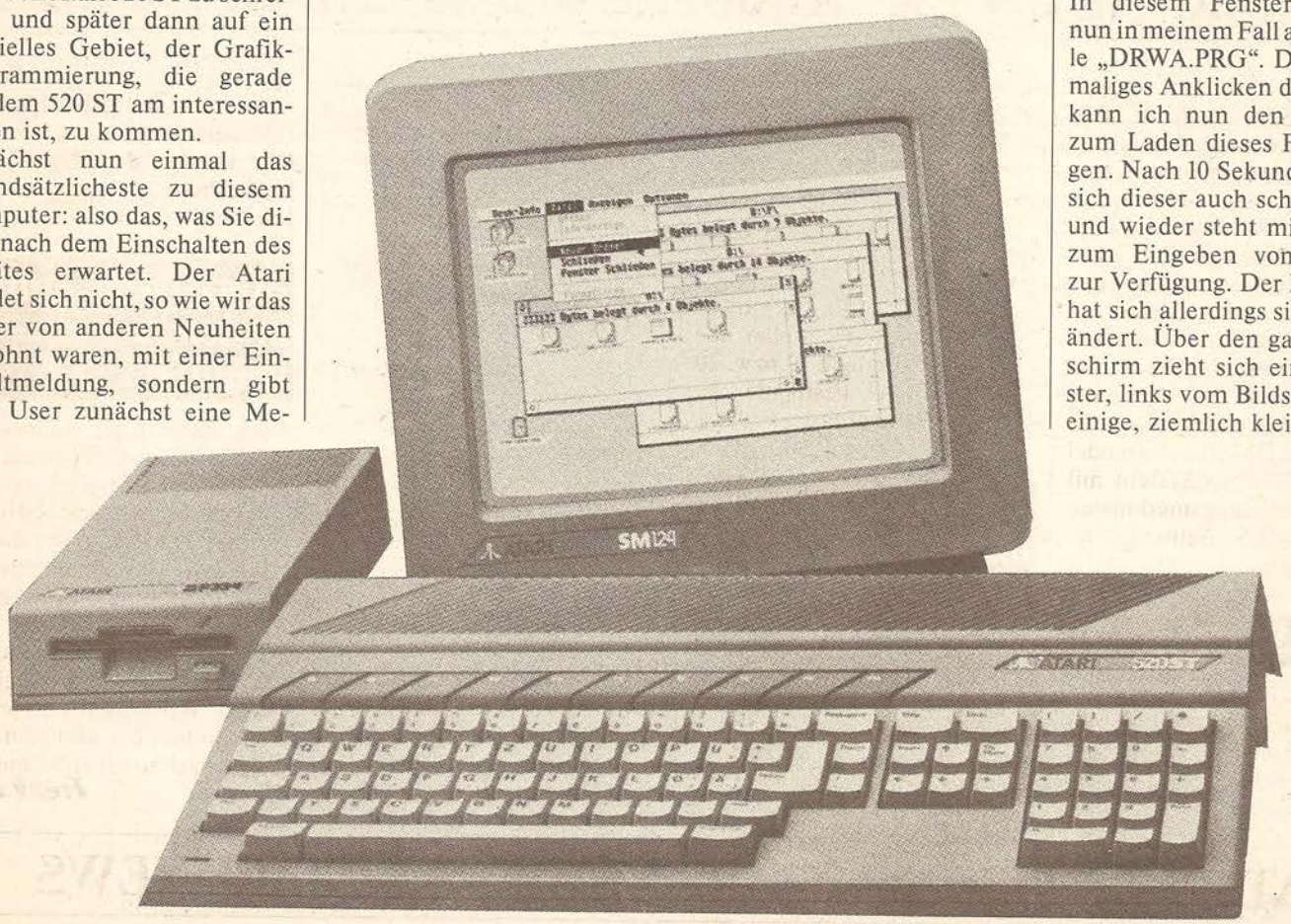
Wenn Sie erwarten nun einen objektiven Testbericht über den Atari 520 ST zu lesen, so muß ich Sie leider enttäuschen. Einen wirklich objektiven Testbericht über dieses relativ neue Gerät zu geben, ist praktisch so gut wie unmöglich, da es so viele verschiedene Möglichkeiten gibt, eine bestimmte Sache zu behandeln, daß dieses dann den Rahmen des Testberichtes sprengen würde. So begnüge ich mich also damit, nur zunächst das Grundsätzlichste über den Atari 520 ST zu schreiben, und später dann auf ein spezielles Gebiet, der Grafikprogrammierung, die gerade bei dem 520 ST am interessantesten ist, zu kommen.

Zunächst nun einmal das Grundsätzlicheste zu diesem Computer: also das, was Sie direkt nach dem Einschalten des Gerätes erwartet. Der Atari meldet sich nicht, so wie wir das früher von anderen Neuheiten gewohnt waren, mit einer Einschaltmeldung, sondern gibt dem User zunächst eine Menueleiste, sowie in der Mitte des Bildschirms eine Art schön gestalteten Cursor aus. Weiterhin signalisiert uns das Floppylaufwerk, dadurch das es anläuft, daß ein Programm, das T(ramiel) O(perating) S(ystem), aufgebootet wird. Nach ca. 30 Sekunden wartet dann dieses TOS darauf, daß mit dem Cursor, der sich inzwischen in einen Pfeil verwandelt hat, eine Eingabe vorgenommen wird. Zusätzlich ist nun in der oberen Bildschirmhälfte ein Window erschienen, das den Inhalt der Diskette anzeigt, auf dem sich auch das TOS befindet. Sie können jetzt mit der Maus auf Ihrem Tisch herumfahren und beobachten, wie sich simultan dazu der Pfeil bewegt.

Anwählen kann man die einzelnen Punkte, indem Sie einfach mit dem Pfeil den entsprechenden Menüpunkt ansteuern, und zwei mal kurz hintereinander diesen Punkt „anklicken". Auch die Windows, die dann neu aufgebaut werden, können manipuliert werden, in dem mit dem Cursor die Window Größe und Lage in den Eckpunkten eingestellt wird.

Ich lege also meine Diskette mit dem neuen Programm „Draw" in das zweite Laufwerk, klicke zweimal den Menuepunkt „Diskettenstation" an, und kann dann beobachten, wie ein zweites Window auf meinem Bildschirm entsteht, das die Directory der Diskette, die sich im zweiten Laufwerk befindet, enthält.

In diesem Fenster erscheint nun in meinem Fall auch das File „DRWA.PRG". Durch zweimaliges Anklicken dieses Files, kann ich nun den Computer zum Laden dieses Files bewegen. Nach 10 Sekunden meldet sich dieser auch schon zurück, und wieder steht mir der Pfeil zum Eingeben von Befehlen zur Verfügung. Der Bildschirm hat sich allerdings sichtbar verändert. Über den ganzen Bildschirm zieht sich eine Art Raster, links vom Bildschirm sind einige, ziemlich klein geratene



Menuepunkte zu sehen, und auch die Menueleiste ist um die Punkte „Font, Page, Arrange, Pattern, Line, Color" erweitert worden, jedoch werde nun auf einige spezielle Operationen, die das Programm, das ganz offensichtlich eine Art Koalapainter darstellt, berichtet. Selbst nur über ein Programm ausführlich zu berichten, würde den Testbericht mehr als vervierfachern.

Punkte zeichnen: Um einen Punkt zu zeichnen, plaziere ich den Cursor zunächst in dem Symbolfeld, das für den Menuepunkt „Punkte zeichnen" darsteht. Danach bin ich in der Lage, frei Hand, oder besser frei Maus, verschiedene Portraits zu erstellen.

Kreis zeichnen: Hierzu fahre ich mit dem Cursor auf den Menuepunkt, der einen Kreis beinhaltet. Ich klicke diesen Kreis nun einmal an, und der Pfeil verwandelt sich in ein kleines Zielkreuz. Nun kann ich durch gleichzeitiges Drücken der Maustaste und Bewegen der Maus, meinen Kreis positionieren, und auch in der Größe variieren.

Dieses sollen nur zwei Punkte von insgesamt acht sein, mit denen es möglich ist, die verschiedensten geometrischen Figuren auf den Bildschirm zu zaubern.

Ich werde mich nun, da ich schon ein schönes Bild auf dem Bildschirm entworfen habe, in die Menueleiste zurückzuziehen, um dort ein bißchen Unordnung zu schaffen. In dieser Menueleiste erwarten mich noch die folgenden Dinge, die ich nun ihn stichwortartig aufzählen möchte:

1. Font: Hier habe ich die Möglichkeit den Bildschirminhalt zu vergrößern, zu verkleinern, oder die Parameter für ein sogenanntes „Ruler", zu deutsch „Lineal" festzulegen. Dies äußert sich dann darin, das an den Rändern eine bestimmte Rastereinteilung vorgenommen wird, die das technische Zeichnen, das durchaus mit diesem Programm möglich ist, stark zu vereinfachen.
2. Arrange: Dieses Untermenue bietet mir die Möglichkeit den Bildschirmausschnitt, den ich sehe, an einen bestimmten Punkt des Gesamtbildschirmes zu legen. Der Gesamtbildschirm beschränkt sich nämlich nicht nur auf eine Bildschirmseite, sondern als ein ca. zehnfaches des eigentlichen Bildschirms, also dessen, was tatsächlich zu sehen ist.
3. Patters: In diesem Menuepunkt habe ich nun die Möglichkeit bestimmte Muster zu wählen, mit dem dann jede umschlossene Fläche ausgefüllt wird. Weiterhin habe ich die Möglichkeit dort zu bestimmen, ob ein von mir gezeichnetes Gebilde einen Schatten werfen soll, oder nicht.
4. Line: Der Menuepunkt LINE, gibt mit die Möglichkeit, die Art der Linie (dick, dünn, gestrichelt, usw.) zu wählen.
5. COLOR: Dieser letzte Menuepunkt ist nun für die Farbwahl zuständig. Dieses ist mir leider nicht möglich, da ich nur über einen Schwarz-Weiß Monitor verfüge.

Ich hoffe, das ich Ihnen mit diesem Testbericht einen kleinen Teil der Möglichkeiten vermitteln konnte, die in diesem Gerät stecken. Erwähnenswert wäre nur noch die Tatsache, daß alle Operationen mit einer solchen Schnelligkeit ablaufen, daß man sie als eingefleischter 64er Anhänger kaum für möglich hält.

* ATARI-NEWS * ATARI-NEWS * ATARI-NEWS *

Neben dem schon weit verbreiteten Atari 520 ST soll nun auch eine etwas abgeschwächte PC-Version unter dem Namen Atari 260 ST auf den Markt kommen. Dieses Modell besitzt die gleichen Eigenschaften wie auch der 520 ST, hat jedoch nur 256 Kilo Bytes RAM Speicher zur Verfügung. Nach den neuesten Informationen der Firma Atari soll das System ohne Maus; Floppy und Monitor für knapp 1000 DM in den Handel gelangen. Für ein System mit solchen Leistungsmerkmalen dürfte das als Sensation zu bezeichnen sein.

Atari-Schlagzeilen

- Nach neuesten Informationen sollen weltweit bereits 30000 Atari ST ausgeliefert worden sein, 4000 davon in der BRD.
- Atari kündigt das Floppylaufwerk Typ 314, mit einer Speicherkapazität von formatiert 720 KB, an. Zusammen mit einer japanischen Firma sind weitere noch leistungsfähigere Laufwerke geplant.
- Atari-Festplatten noch nicht geklärt! Ein Hersteller für die angekündigte 10 bzw. 20 Mega Byte Festplatte ist noch nicht gefunden.

Atari ST Software im Kommen!

Eines der Schwachpunkte des Atari 520 ST scheint überwunden zu sein. Einige Softwarehäuser stellen ihre ersten Programme vor.

Metacombo kündigt an:
- Pascal-Compiler
- Makro Assembler nach Motorola Standard
- Lisb Interpreter
- BCPL Compiler
- Lattice C Compiler
- IBM PC Konverter

Bereits lieferbare Programme:
- UCSD Pascal
- TDI Modula -2/ST

Die durch Schneider bekannte Firma Hisoft plant:
- Programmiersprache C
- Pascal
- DEVPAC

Das Haus Kuma-Computer kündigt an:
- K-Seka, Editor/Assembler
- K-Data, Dateiverwaltung
- K-Word, Textverarbeitung
- K-Komm, Kommunikationssoftware

Da weltweit auf Hochtouren an ST-Software gearbeitet wird, dürfte es nur eine Frage der Zeit sein bis die ersten Action Games über den ST-Bildschirm flimmern.

Wir werden den Atari-Markt beobachten und Sie auf dem Laufenden halten.

Atari-Basic kommt in den Handel!

Nach zahlreichen „Probe-Basic" Versionen kommt nun endlich eine brauchbare Version in einer größeren Stückzahl in den Handel. Diese Version umfaßt 150 KByte Maschinencode und dürfte damit die längste Basic-Version auf dem Markt sein. Was diese Version genau leistet, konnten wir noch nicht erfahren. Wir hoffen dies jedoch bald in einer der nächsten Ausgaben nachholen zu können.

Frank Brall

* ATARI-NEWS * ATARI-NEWS * ATARI-NEWS *

CPU-Buchkritik: Mailboxführer - Mailboxführer - Mailboxführer

Autor: Bruno Hurth; Sybex-Verlag Düsseldorf; Preis: 14,80 DM; ISBN 3-88745-620-3

Anscheinend kann man fast alles zu Geld machen. Auch ein paar ausgedruckte Menüs ...
Aber Scherz beiseite, dieser Mailboxführer soll für DFÜ-Freaks und auch Anfänger interessant sein. So steht es auf dem Umschlag. Für einige wenige Rubriken in diesem Buch stimmt das wohl auch.
Zunächst zum Hauptpunkt, der Übersicht über die Mailboxen. Hierbei handelt es sich um eine Anzahl von ausgedruckten Menüs, teilweise auch ein wenig mehr. Außerdem findet der Leser zu Beginn jeweils die Angabe der Parameter, Betriebszeiten, Zeitlimit und Timeout.
Leider sind einige interessante Boxen nicht aufgeführt, so z. B. das Bremer Mailboxsystem (0421-425193, Sysop Gerold, 24 h). Leider stimmen bei anderen, z. B. der Tornado (Hamburg) die Parameter nicht, die richtigen Parameter sind 8 Daten-, 1 Stopbit, keine Parität (es dürfte aber mit der angegebenen Einstellung 8-2-keine auch gehen).
Natürlich ist es teurer, alle Boxen anzurufen und sich die Menüs senden zu lassen, jedoch wird es teilweise trotzdem erforderlich sein. So haben etliche Boxen inzwischen Hobby-Ecken, für die meistgefragten Rechner extra. Welche Typen aber in welcher Box zu finden sind, geht aus diesen Ausdrukken nur andeutungsweise, wenn überhaupt hervor.
Auch die Angaben zum System, mit dem die Box gefahren wird, sind unvollständig. So hat Tedas z. B. einen Epson QX 10, und das kann auch in der Box erfahren werden. Dem Führer ist zu entnehmen, daß hierzu keine Angaben vorliegen.
Insgesamt bleibt für diesen Teil zu sagen, daß die Angaben unnötigerweise unvollständig bzw. falsch sind, wobei die Fehler jedoch keine größeren Schwierigkeiten verursachen dürften.
Zu Beginn wurden für den von seiner Rechnung geschockten DFÜ-Fan einige Telefonspartips zusammengetragen. Einer davon ist offensichtlich kaum sinnvoll: das Überspringen von Nachrichten mit CTRL-C bringt kaum einen Zeitvorteil; die Dateien sind in der Regel seriell angelegt und müssen auf jeden Fall zuende gelesen werden, auch wenn der Rest nicht gesendet wird.
Es ist auf keinen Fall sinnvoll, grundsätzlich alle abgerufenen Daten abzuspeichern! Leider hat man dann nämlich den Speicher voll mit Sternchen, Hinweisen auf Werbung, die zufällig für einen anderen Computer ist, Müll (einige User sind ganz groß im Produzieren von unschönen Bemerkungen!); vor Beginn der MSG weiß man leider nicht, was darin steht. Aber warum bitte soll das auch noch abgespeichert werden? Sinnvoller ist es hier, bei den Boxen, die man öfter anrufen will, die Menüs einmal komplett (= voll ausgeschrieben) abzurufen und auszudrucken. Dann schneidet man sich das aus und klebt es in Computerreichweite an die Wand oder macht „Hausaufgaben" und lernt es auswendig. Dann kann man auf „Experte" umschalten, so daß nur noch kurze Eingabehinweise oder gar keine Menüs mehr erscheinen.
Der Tip mit dem Billigtarif hat einen wesentlichen Haken:
Zu bedauern sind die Wählscheibentelefonbesitzer ohne Wahlwiederholung!
Gut ist der Tip mit Berlin, wer weiß das schon, daß es nach dorthin billiger ist!
Ganz kurz werden in diesem Buch dann noch folgende Kapitel geboten:

- Wie bekomme ich eine NUI für DATEX-P?
- Übersicht über DATEX-P-Boxen
- Übertragungsfrequenzen
- Anschlußschemata für einige wenige Rechner
- Übersicht über die im Handel befindlichen Akustikkoppler
- Eine kurze Mailboxliste
- Ein kurzes DFÜ-Wörterbuch

Der „unentbehrliche Begleiter für Sie" - so heißt es auf dem Buchumschlag - ist dieser Mailboxführer nicht. Es ist natürlich nicht ganz uninteressant, auch mal andere Menüs zu lesen, doch man kommt nach einigen Versuchen sehr gut ohne aus. Etliche Informationen findet man ja auch in den diversen Zeitschriften, von denen fast jeder sicherlich die eine oder andere wenigstens auf Interessantes hin durchsieht.

Erika Hölscher

BDVI sucht noch gleichgesinnte Hobbyfreunde

Die BDVI- fördert und pflegt das gemeinsame Interesse an Bildschirmtext, Datenfernübertragung und Video-Bildplattentechnik.
Wir sind eine Gemeinschaft mit einer gleichberechtigten Arbeitsgruppe. Alle Mitglieder können dort mitgestalten. Alles wird gemeinsam verwaltet. Unsere vielseitigen Interessen ermöglichen es, uns dort wo wir gern gesehen sind und es uns gefällt, Clubtreffen durchzuführen. Unsere Aufgaben sind praktische Ausübung von BTX im Anbieterbereich und Datenfernübertragung. Wir bestimmen mit einfacher Mehrheit.
Jedes Mitglied hat die Möglichkeit, eigene BTX-Seiten über eine Anbieterstation ins System zu geben und die Benutzung unserer eigenen Mailbox mit Paßwort.
Alle Vorschläge werden besprochen. Kein Vorschlag ist zu gering. Anfragen sind zu richten an die Arbeitsgruppe. Eingaben über die Mailbox werden vorgetragen und beantwortet. Wir wollen unbescholtene Bürger aller Altersklassen als Mitglieder gewinnen. Unsachliche Verhaltensweisen werden gemeinsam abgewiesen. Wir wollen unser interessantes Hobby ausbauen, Erfahrungen sammeln, miteinander austauschen und immer auf dem neuesten Stand sein. Wenn unsere Zeit und Kasse es erlauben, werden wir einen Rundbrief- oder Rundtonband-Service einrichten, damit auch Behinderte und verhinderte Hobbyfreunde an unserer Tätigkeit teilhaben können. Behinderte Hobbyfreunde werden, wenn es gewünscht wird, nach vorheriger Terminabsprache von einigen Mitgliedern besucht.
Neue Ideen gegenüber sind wir stets aufgeschlossen und bitten ggf. um Mitteilung.
Der Unkostenbeitrag von DM 36,-- für das ganze Jahr wird bewußt so niedrig gehalten, um jeden Interessenten die Mitgliedschaft zu ermöglichen.

Unsere Anschrift:
BDVI-Mecklenburgerstr. 8
4150 Krefeld 11
Unser Konto:
Sparkasse
4150 Krefeld 11
Kto.-Nr. 60 02 90 55
BDV-Interessengemeinschaft
Unser BTX-Programm:
BTX-Seite ab
92 01 63 01 20
Unsere Mailbox:
Modem-02 151 47 65 67
Telefon: v.10.00 - 20.00
0 21 51 / 47 42 39

Wir haben zwei Super-Chancen, die man sich nicht entgehen lassen sollte!

Dies ist was für scharfe Rechner!

CPU Bestellkarte-Softwareservice

Alle im Heft abgedruckten Programme können als zusätzlicher Service über den Verlag bezogen werden.
(Ausland nur gegen Vorkasse)

Die Zustellung erfolgt: gegen **Vorkasse** ☐ oder Inland per **Nachnahme** ☐ + Versandkosten

innerhalb von 1 Woche

Entnehmen Sie bitte aus unseren Preislisten die notwendigen Angaben für Ihre Bestellung:

Bitte liefern Sie mir

☐ Cassette für _____ (System) ☐ Anzahl

☐ Diskette für _____ (System) ☐ Anzahl

zum Preis von gesamt _____ DM

Name/Vorname: Straße, Nr.:

PLZ/Ort: Datum, Unterschrift

Abonnements:

wer jetzt in's Computerleben einsteigen will, sollte nicht überlegen: ein Abonnement ist immer eine lohnende Sache!

Ausschneiden und einsenden an:

Verlagsunion
Friedrich-Bergius-Straße 20
Postfach 57 07
6200 Wiesbaden

Bestellgutschein

Ja, ich möchte ins Computerleben einsteigen und bestelle deshalb:

☐ Jahresabonnement **CPU** (6 Hefte) 60,- DM

☐ Jahresabonnement **HC und Computronic** (12 Hefte) 60,- DM

☐ Jahresabonnement **Kombi: HC, Computronic und CPU** (18 Hefte) 85,- DM
(Ausland: siehe Impressum)

Name/Vorname

Straße, Nr. PLZ, Ort

Ich wünsche folgende Zahlungsweise:

☐ Bargeldlos durch Bankeinzug _____ Bankleitzahl

Geldinstitut Konto-Nr.

☐ gegen Rechnung _____ Datum, Unterschrift

Abonnements-Kündigungen:
6 Wochen vor Ablauf des Jahresabonnements.
Außerdem kann ich diese Bestellung innerhalb von 10 Tagen beim Verlag widerrufen!
Zur Wahrung der Frist genügt die rechtzeitige Absendung des Widerrufs!

_____ Datum, Unterschrift

Bestellen Sie telefonisch: ☎CPU-Software-Service: 0 56 51 / 4 06 43 oder 4 06 93

Programm	Kassette	Diskette	Bestell-Nr.	System
Multi-Key/S-Tool/ Interrupt-Programm	16,-	21,-	UC 2/85	*Commodore 64*
Spritehilfe/Diskloader/Directory	-	21,-	UC 3.1/85	
Tape-Directory/Asmon/Data-Generator	15,-	21,-	UC 3.2/85	
Fast-Load	10,-	-	UC 4/85	
Diskmonitor/Zeichensatz	-	20,-	UC 5/85	
Reassembler/Maskengenerator	15,-	20,-	UC 6/85	
Del 64/Treiberprogramm für 1526/MPS 802 Decelerator	14,-	19,-	UC 7/85	
Supertape-Directory/Renew/ Controllport-Abfrage	15,-	-	UC 8/85	
Video-Utilitys/ Multi-Basic V3/Bildeditor	17,-	23,-	UC 9/85	
Better Basic/Autonumber/ Hardcopy/Terminalprogramm	17,-	23,-	UC 10/85	
Speicher-Oszilloskop/ Makro-Assembler/Change Type	49,-	54,-	UC 1/86	
Zeichendesigner	14,-	-	UO 8/85	*Commodore C 16/C 116*
OLD-Routine/Merge-Routine/ Change Type	14,-	19,-	UO 1/86	
Autorennen	11,-	-	US 3/85	*Schneider CPC 464*
Universal Datei V1	9,-	-	US 4/85	
Computerschrift/Symbol Swap	10,-	-	US 5/85	
Keyboard Toolkit/Farbdekodierung	11,-	-	US 6.1/85	
Aladin	9,-	-	US 6.2/85	
Painter/Box-Befehl	14,-	24,-	US 7/85	
Maschinensprache-Monitor/ Disk-Hilfe	15,-	25,-	US 8/85	
Basic/Maschinenkit	12,-	24,-	US 9/85	
Back-up/Directory/ Ellipse/Disk	-	27,-	US 10/85	
Data-Generator/Grafik Toolkit ASCII-List (Turbo Pascal)	15,-	25,-	US 1/86	
Olympics	11,-	-	UT 2/85	*TI-99*
Funktionsanalyse/Hardcopy	11,-	-	UT 3/85	
Invaded by the Empire	11,-	-	UT 4/85	
Zeichendefinition	14,-	-	UT 8/85	
Superfile	11,-	-	UT 9/85	
Basic-Erweiterung	11,-	-	UT 1/86	

Programm	Kassette	Diskette	Bestell-Nr.	System
Mauerflucht/Lunar-Simulation	13,-	19,-	UI 2/85	*Atari*
RAM-Tester/Cherry Harry	13,-	-	UI 3/85	
Crazy Egon	13,-	19,-	UI 5/85	
Monitorprogramm	11,-	-	UI 6/85	
Zeicheneditor	-	19,-	UI 7/85	
Kopfzeilengenerator/ Lohnabrechnung	-	22,-	UI 8/85	
Grafikausgabe/Error-Helper	-	18,-	UI 9/85	
Translator/Zeichendesigner	14,-	20,-	UI 10/85	
Sektor Kopierer/Epson-Konverter/ Translator	17,-	22,-	UI 1/86	
Mathematische Ableitungen	nur auf Anfrage!!!			*Atari 520 ST*
Zeichensatzkonverter	10,-	-	UM 3/85	*MSX*
Pipeline	13,-	-	UM 4/85	
Zirkus	13,-	-	UM 5/85	
Alien Attack	13,-	-	UM 7/85	Disketten nur im Format 5 1/4 Zoll erhältlich!
Anti-Rom	12,-	-	UM 8/85	
Typer 60	19,-	25,-	UM 9/85	
Disketten-Monitor	-	25,-	UM 1/86	
Berechnung von aktiven Filter-schaltungen/Screen-Save/ Z-Konverter/Sprite-Positionen/ Joystickabfrage	17,-	-	UM 10/85	
Panik Sam/Colour Change/Clock	13,-	-	UZ 2/85	*ZX-Spectrum*
MC-Routinen/Data-Base/Pixel	11,-	-	UZ 3/85	
Reversi/UDG-Editor	15,-	-	UZ 4/85	
Star Basic	13,-	-	UZ 5/85	
Fill-Programm/Copy-it	14,-	-	UZ 6/85	
Quassi	13,-	-	UZ 7/85	
Intersound/Balken	14,-	-	UZ 8/85	
Trace-Single Step/ Variable List/X-Basic	16,-	-	UZ 9/85	
Disassembler/Ohmsches Gesetz/	17,-	-	UZ 10/85	
Zeilen Kill/Grafik Compiler	17,-	-	UZ 1/86	
Asteroids	-	10,-	UA 2/85	*Apple*
Dateiverwaltung	-	18,-	UA 3/85	
Digiter/Reset-Saver	-	18,-	UA 4/85	
Assembler-Monitor	-	17,-	UA 5/85	
Disk Check IC/Dos Change	-	18,-	UA 7/85	
Hello	-	17,-	UA 7/85	
As-Editor	-	18,-	UA 8/85	
Digtal-Sound-Editor/ Basic-Befehle	-	18,-	UA 9/85	
Big Spender	-	19,-	UA 10/85	
RGS-Synthesizer	-	19,-	UA 1/86	
3D-Rotation	17,-	-	UD 1/86	*Dragon*

Verkäufe

●●● **ATARI 400 - 130** ●●●
Endlich lieferbar: Charakter-Analyse/Horoskop (Disk), 10 Seiten Auswertung u. Farbgraphik!! DM 59,–. NN od. VK. Info gratis. T. Jenczyk, PF. 65, 3250 Hameln

Schneider CPC 464/6128, 15 Anwender-Progr., Mathem., Physik, Chemie, Nirtsch. usw. Kass. 30 DM, Disk. 40 DM in Scheinen o. Scheck, K. Bienek, Klosterstr. 31, 1000 Berlin 20

✦✦✦✦✦✦✦✦✦✦✦✦✦✦✦✦
✦ **SCHNEIDERCPC464/664** ✦
✦ Univers. Ein/Ausgabe-In- ✦
✦ terface 32 Bit TTL. Zum ✦
✦ Anschl. vielseit. Hardware ✦
✦ a. CPC. z.B. A/D Wandler, ✦
✦ Relais, Roboter, Schul-Ei- ✦
✦ senbahn-Discoelectr. od. ✦
✦ M&L Zubehör. Ausf. Be- ✦
✦ dien.-Anl. Eignes Netzteil. ✦
✦ Programmierb. in Basic ✦
✦ od. Asemml. Interupt- ✦
✦ taugl.* DM 198 EPROM- ✦
✦ PROGRAMMIERGERÄT ✦
✦ an E/A-Interface an- ✦
✦ schließb. f. 2716-27128, ✦
✦ 2516-2564 m. komfort. ✦
✦ Bedienersoftw. 28pol. ✦
✦ Textoolsock. vorber. f. ✦
✦ 27256. Maschinenspr.- ✦
✦ Monitor. Lesen, Leert. ✦
✦ Programm. Vergleich. v. ✦
✦ Eprom's * DM 178 M&L- ✦
✦ Maximini&Lichter, ✦
✦ Hauptstr. 1a, 5511 Manne- ✦
✦ bach, T. 0 65 81 / 35 35 ✦
✦✦✦✦✦✦✦✦✦✦✦✦✦✦✦✦

Verk. Top TI 99/4A + Cass. Rec. + P-Box kompl. mit int. Contr., Disk, 32 K, RS 232 + Joyst. + Ex Bas + Ed. Assem. + Sprachsynt, + 8 Mod. Preis = VHS. M. Sandmann, Im Brauke 5, 4630 Bochum, Tel. 38 25 96

An alle TI-99/4A-Anwender! Kassette mit 4 Info-Progr. vollgepackt mit tollen Basic und ED/Ass.-Progr. DM 5,– in Umschl. Bei Kauf Vergütung. An Karbach, Remscheiderstraße 18, 5650 Solingen 1

SENSATIONELL: Zusatzlaufwerke für Schneider **CPC-464** und **CPC-664.** Im Gehäuse und mit allen Kabeln 5 1/4" / 40 Track / einseitig **549,– DM.** 3" / 40 Track / einseitig **399,– DM. KDB-COMPUTER-VERSAND,** Kornstraße 28, 5800 Hagen 7, Tel. 0 23 31 / 40 06 01

G&S Computerservice, Rudi Gutemann u. Dieter Schulz, Soft- u. Hardware C-64. Unser Angebot: Farbb. Epson 80, 801, 802 je 15,95, Disketten 3. 50 St. 10er Box 39,50, Abdeckhaube 12,25, Lüfter 37,90, Box 85 Disk. m. Schloß 35,50, Diskettenloch. 12,95 Eprombrenner kpl. 159,00. D. Schulz, Funkenstr. 42, 5020 Frechen, Tel.: 0 22 34 / 5 78 48. Katalog kostenlos.

✦✦✦ **TOP ANGEBOT** ✦✦✦
✦ ✦
✦ Verkaufe C 16 + Data- ✦
✦ sette + 2 neue, kaum ✦
✦ gebrauchte Joysticks, ✦
✦ NP 580 DM für 190 DM, ✦
✦ Tel. 0 70 22 / 3 32 45 ✦
✦ ✦
✦✦✦✦✦✦✦✦✦✦✦✦✦✦✦✦

64 K RAM MSX Tape Back-Up, völlig MC 1200/2400 Baud, Motor, Rename etc. Bytes left, mehr als 60 k benutzbar, sende Scheck (Euro) 35 DM an van Kinderen, Haagbeukhof 249, 3355 AG Papendrecht, Nederland

Hallo TI 99/4 A Freaksb 32 k RAM Speichererweiterung extern als BS: 1950 DM o. FG: 240 DM, was sagt Ihr nun? Info u. Bestellung an H. Ayasse, Schöckengasse 5, 7253 Renningen 2

Elektronik Software ✦ Spectrum, Transistor Vorstufenberechng. 16 K ✦ (Heft 1 Fundgrube) 30 DM ✦ Graphikprogramm-Schaltung 48 K 40 DM ✦ Democass. 3 DM ✦ K. Thiesen, Rathausstraße 70, 5410 Höhr-Grenz

✦✦✦✦✦✦✦✦✦✦✦✦✦

Alpha-Datek V 1.0 99'er Spitzen Software. Ihre Profi Datenbank f. Ihren TI 99/4A. Ein Super Progr. mit 40 Zeichen pro Zeile, SUPER schnelles Sortieren und echte Kleinb. durch E/A Unterprogr. und vieles mehr f. wenig Geld. Info by U. Brüsseler ab 20 Uhr, 4050 Mönchengladbach 3, Wetschewell 125 d, T. 0 21 66 / 5 34 57

Kaufgesuche

?? Help ?? Hilfe ?? Help ?? Suche dringend gebrachten C-64 mit Floppy-Laufwerk. Angebote an H. Kästle, Marktstraße, 3440 Eschwege oder telef. unter 0 56 51 / 5 06 85

Suche käuflich Zeitschriften. Homecomputer 3/83 und 1–7/84, außerdem CPU 1–6/84 + 8/84. Listings müssen lesbar sein. – Chiffre 1 –

Kontakte

Der Besitzer des „1. Homecomputers in dt. Gefängnissen" sucht netten Menschen, der ihm Zubehör (z. B. Ex. Basic) schenkt! Anschrift: Jens Schiffhauer, Tändersche Str. 50, 3550 Hameln, JVA (Haus 5) – Danke! –

Club der Toten Geesthacht für VC 20 und TI-99/4A Club-Aktivitäten: Software-Austausch – 5jährlich Clubinfo – Club-Treffen, Einkaufsgemeinschaft etc. Konfigurationen: VC-20-Konsole, Datasette, Drucker, 32/37 KB, Erweiterung, Joystick, Software. TI 99/4A-Konsole, Kassettenrecorder, Extended Basic, Mini Memory, 32 KB Erweiterung, Centronics Schnittstelle, Joystick, Sprachsynthesizer, Software (auch Module). Clubinfo gegen Rückporto, Clubbeitrag 1,– DM (dann Zeitschrift kostenlos). Anschrift: H. Wolf, Club der Toten, Am Hang 5, 2054 Geesthacht

Verkäufe

Herzlichen Glückwunsch!!
Zu 1 Jahr COMBI-CLUB und über 210 Mitgliedern gratulieren die Leiter der User-Gruppen Nord, Süd und Westfalen!

GESTRICHEN . . .
haben wir fast alle Softwarepreise (z. B. CYRUS II 3-D Schach DM 41,90) u. das Angebot auf über 140 Artikel erweitert! Die Mitgliedschaft ist immer noch für 20,––/Halbjahr zu haben!! COMPI-CLUB, J. Heise, A.d. Linde 8, 5226 Reichshof.

SOFT-TEAM-BERLIN
Software und Zubehör für
C-64 Atari Schneider
Spectrum MSX
Über 2000 Programme immer auf Lager.
Ständig die neuesten Programme aus England und den USA.
Gratiskatalog anfordern:
Anton Peter & Partner, Kamminerstraße 9
1000 Berlin 10, Telefon 453 27 11

Autoren-Wettbewerb in „Tips u. Tricks von CPU:“ Der Leser wird mitentscheiden, wer gewinnt!

Das CPU-Team war wieder „aus dem Häuschen“: Mit solch' einer Resonanz durch Sie, den Leser, hatten wir nicht gerechnet. Jetzt geht's in die dritte Runde. Und: Der Leser ist aufgerufen mitzuwirken!

Tolle Gewinne warten auf Ihre Gewinner: Insgesamt **600 Deutsche Mark** *werden jeden Doppelmonat für die besten Autoren-Beiträge „an den Mann gebracht“. Die Wettbewerbs-Kriterien: Der Wettbewerb gilt für die besten Einsendungen in den Rubriken*

1. User (maximal drei Seiten sollten bei uns „ankommen“)

2. Utility (hier sollten es max. zwei Seiten sein)

3. Tips & Tricks (hier maximal eine Seite)

Der Leser – als Juror – wird mitentscheiden!

Das Team der CPU wird dann die besten Einsendungen und Tips wie folgt honorieren:

für den besten „User-Text“ = 300 Mark,

für die beste „Utility“ = 200 Mark und

für die besten Tips & Tricks = 100 Mark

Sie sehen also, Mitmachen lohnt sich! Nur, wer mitwirkt, kann gewinnen.

Und: Mehr noch – der „Spitzen-Autor“ wird in der nächstfolgenden Ausgabe der CPU vorgestellt.

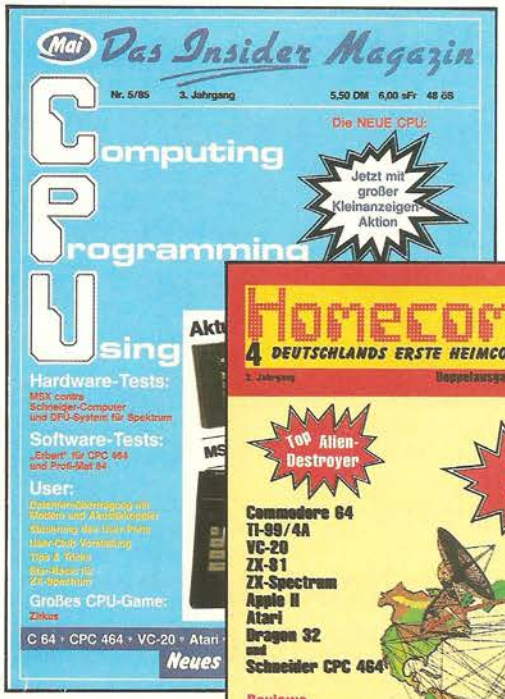
Wenn Sie sich an diesem Wettbewerb beteiligen wollen, übersenden Sie uns bitte folgende Unterlagen:

1. **Ausführliche Beschreibung des eingesandten Beitrages**
2. **Datenträger (Kassette oder Diskette)**
3. **Komplettes Listing (wenn zur Beschreibung erforderlich)**
4. **Für die Autorenvorstellung: Infos zur eigenen Person (kurzer Lebenslauf, Hobbies etc.) + Foto**
5. **Frankierten Freiumschlag (mind. 1,40 DM)**

All dies senden Sie bitte an den TRONIC-Verlag, Postfach, 3444 Wehretal 1.

Also, mitmachen, mitwirken und mitgewinnen! CPU – das Magazin mit Pfiff!

Ihr Erfolgsprogramm – aus dem Tronic-Verlag

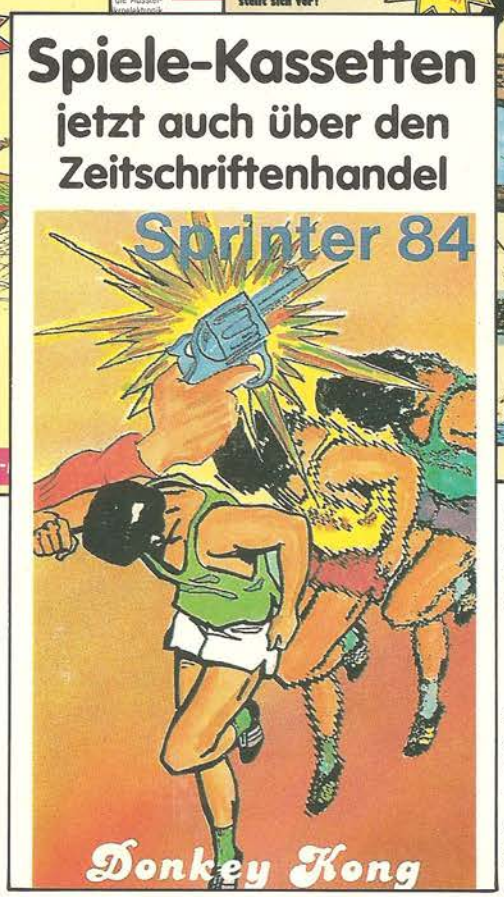


CPU
zweimonatlich
Format 21 x 28 cm
6,– DM

Compute mit
monatlich
Format 21 x 28 cm
2,80 DM

Homecomputer
zweimonatlich
Format 21 x 28 cm
6,– DM

Computronic
zweimonatlich
Format 21 x 28 cm
6,50 DM



Spiele-Kassetten:
Commodore 64
VC-20, Atari,
TI-99, ZX-Spectrum,
Laser 2001,
Schneider CPC 464
Je 19,90 DM

Spitzenprodukte

für den wachsenden Markt der Heimcomputer-Anwender

Informationen: Tronic-Verlag, Landstraße 29
Postfach 41, 3444 Wehretal 1, Ruf (0 56 51) 4 06 93
Vertrieb: Verlagsunion, Friedrich-Bergius-Straße 20, 6200 Wiesbaden, Ruf (0 61 21) 26 60